

TP vision industrielle

SOMMAIRE

QU'EST-CE QUE LA VISION INDUSTRIELLE ?	7
1. OBJECTIF ET SUPPORTS	7
2. INTRODUCTION A L'ANALYSE D'IMAGES NUMERIQUES	8
2.1. Formation, traitement et analyse d'images	8
2.2. Analyse quantitative de structures	9
2.3. Reconnaissance de formes	10
3. ACQUISITION D'UNE IMAGE NUMERIQUE PAR CAMERAS CCD (CHARGE COUPLED DEVICE)	11
3.1. Capteurs linéaires	11
3.2. Capteurs matricielles	12
4. PRESENTATION DE LA PLATE-FORME DE VISION INDUSTRIELLE	15
4.1. Introduction	15
4.2. Description	15
4.3. Moyens matériels et logiciels	16
TP INSPECTOR : ACQUISITION ET ANALYSE D'IMAGES AVEC LE LOGICIEL MATROX INSPECTOR	17
1. PRISE EN MAIN DU LOGICIEL INSPECTOR	17
1.1. Acquisition d'images	17
1.2. Histogramme	18
1.3. Négatif d'une image	18
1.4. Manipulation d'histogramme	18
1.5. Quantification	21
1.6. Opérations morphologiques (voir annexes)	22
2. CHAINE DE TRAITEMENT	25
2.1. Acquérir les images	25
2.2. Les traiter par la chaîne mise en place	25

TP MIL : ANALYSE D'IMAGES AVEC LA LIBRAIRIE MIL (MATROX IMAGING LIBRARY)	27
1. INTRODUCTION A LA LIBRAIRIE MIL	27
1.1. Les différentes étapes afin d'inclure une librairie dans un projet	27
1.1.1. Création d'un espace de travail (Workspace) sous Visual Studio	27
1.1.2. Insertion de la librairie MIL sous Visual Studio	30
1.2. Mode d'emploi d'apprentissage de la librairie MIL	31
2. LECTURE D'UNE IMAGE A PARTIR DE MIL	33
2.1. Récupérer un code déjà existant	33
2.2. Analyse du code proposé	33
3. MANIPULATION D'HISTOGRAMME	34
3.1. Rendre le nouveau projet prioritaire	34
3.2. Récupérer un code déjà existant	34
3.3. Analyse du code proposé	34
3.4. Manipulation d'un histogramme	35
3.5. Seuillage (ou binarisation) de l'image	35
4. FILTRES MORPHOLOGIQUES	36
4.1. Méthode de bouchage de trous	36
4.2. Méthode de séparation et dénombrement de pièces	37
5. METROLOGIE : METHODE DE MESURE DE CARACTERISTIQUES D'OBJETS	38
5.1. Méthode de détection de contours	38
6. CHALLENGE	39
ANNEXES	40
ANNEXE 1 : METHODES EN TRAITEMENT D'IMAGES NUMERIQUES	41
6.1. Acquisition	41
6.2. Représentation d'une d'image numérique	43
6.3. Transformations d'images	43
6.3.1. Opérations entre images	43
6.3.2. Transformations point par point	45
6.3.3. Filtres linéaires par convolution	46
6.4. Mesures	46

6.4.1.	Fenêtre de mesure	46
6.4.2.	Paramètres de forme	47
6.4.3.	Paramètres de texture	48

ANNEXE 2 : FILTRES DE MORPHOLOGIE MATHEMATIQUE 50

6.5.	Principe de filtrage en morphologie mathématique	50
6.6.	Erosion et dilatation	51
6.7.	Ouverture et fermeture	51
6.8.	Squelette	53
6.9.	Transformations géodésiques	53

QU'EST-CE QUE LA VISION INDUSTRIELLE ?

Dans les productions à haute cadence, le souci constant d'amélioration de la qualité et la recherche de gain économique poussent de plus en plus les industriels à automatiser les moyens de production. La vision industrielle est une réponse à ces préoccupations pour les opérations de contrôles de la production. En effet les machines de vision industrielle permettent un contrôle de la production à haute cadence et assurent une bonne répétabilité du contrôle. Les systèmes de vision sont également énormément utilisés pour gérer des flux d'objets, par exemple pour la lecture optique des codes barres ou le tri d'objets (pièces mécaniques, produits alimentaires) par couleurs différentes avant emballage. Enfin la vision industrielle peut être un moyen de guidage pour un système mobile autonome (comme un robot) lorsque ses mouvements ne peuvent pas être déterminés par avance, comme par exemple la préhension d'objets sur un tapis roulant. Une caméra est alors embarquée sur la tête du robot et permet le positionnement de celui-ci au point désiré.

En étudiant l'évolution des technologies, on peut se rendre compte que les systèmes de vision industrielle sont de plus en plus présents dans notre environnement. Dans l'univers du multimédia, la vision industrielle est présente sous forme de système embarqués temps réel s'appuyant sur des traitements d'image et de reconnaissance de formes. Dans le monde de l'entreprise, de plus en plus de caméras dites « intelligentes », c'est à dire permettant à la fois d'acquérir une image, d'extraire l'information utile et de la transmettre, s'imposent comme des outils efficaces et rentables. La présence de plus en plus forte de la vision industrielle dans les entreprises des pays occidentaux s'explique finalement facilement par la nécessité d'être compétitif et donc de rendre un certain nombre de tâches automatiques, contrôlées et régulées.

1. OBJECTIF ET SUPPORTS

Ainsi, l'objectif de cette "plate-forme a pour but de vous fournir une auto formation de huit heures à la vision industrielle. Pour ce faire, nous avons mis en place un support de formation constitué de deux TP de quatre heures ayant des buts bien définis :

- former les étudiants hors option TdSI aux bases de la vision industrielle sous la forme de TP transverses ;
- mettre en pratique le cours existant de "traitement d'images" pour les étudiants de l'option TdSI sous la forme de TP approfondis.

Afin d'être au plus près de l'actualité industrielle en terme de vision, vous allez vous former à des outils de vision industrielle actuels au travers d'un logiciel d'analyse d'image INSPECTOR et d'une librairie de développement d'outils d'analyse d'image en langage de programmation C appelée MIL. De plus, afin de faciliter la rotation des TP, les deux TP peuvent être abordé de façon indépendante. Afin d'acquérir les connaissances nécessaires au bon déroulement des TP, nous avons réalisé des annexes qui détaillent des parties importantes de la formation. Ainsi une même annexe peut servir pour les deux TP.

Avant de décrire plus précisément le contenu de chaque TP, nous débutons cette formation par une courte introduction à l'analyse d'images numériques (section 2) et au principe d'acquisition d'une image numérique par caméra CCD, caméras très utilisées en industrie (section 3). Finalement un descriptif détaillé de chaque TP est donné en section 4.

2. INTRODUCTION A L'ANALYSE D'IMAGES NUMERIQUES

Source : *Technique de l'ingénieur : analyse d'images* (<http://www.techniques-ingenieur.fr>)

L'analyse d'images numériques continue de se développer et s'implante dans de plus en plus de secteurs industriels. Méthode générale aux applications diversifiées, elle permet d'extraire de façon objective, précise et souvent automatique l'information contenue dans une image. Elle conduit à des mesures ou à des reconnaissances de formes, d'objets ou de structures à partir de leurs images. Elle trouve des applications dans de nombreux domaines industriels: contrôle industriel, science des matériaux, robotique, science de la vie, où elle permet des automatisations, des mesures sans manipulation directe sur les objets. Cette partie a pour but de donner un aperçu le plus complet possible de l'analyse d'images. La première partie présente les principales applications de l'analyse d'images. La seconde partie, plus méthodologique, explique les étapes successives d'un programme d'analyse : acquisition, segmentation, mesures et interprétation des résultats.

2.1. Formation, traitement et analyse d'images

Le domaine de l'imagerie est très vaste. Il désigne toutes les méthodes qui font intervenir « une image ».

Formation

Le domaine de l'imagerie le plus large se situe en amont. C'est celui de la formation et de la synthèse d'images. Les systèmes de formation d'images que nous utilisons le plus souvent travaillent dans le visible, en lumière du jour ou artificielle ; les appareils photographiques, cinématographiques, vidéo en sont des exemples. D'autres systèmes fonctionnent à des longueurs d'ondes électromagnétiques plus courtes (ultraviolet, rayons X) ou plus longues (infrarouge). Enfin, d'autres appareillages utilisent les propriétés des particules chargées (microscope électronique, analyseur ionique). Le développement des techniques permet à l'image de devenir de plus en plus une source d'information et non pas uniquement un simple outil de circulation et de stockage d'informations. Une image est une représentation bidimensionnelle d'un objet accessible ou non à l'observation visuelle, et qui est obtenue à un instant et à un grossissement donnés.

Traitement

Les méthodes de traitement d'images désignent les outils utilisés pour transformer une image numérique en une nouvelle image, ou même combiner plusieurs images entre elles. Elles permettent de compresser l'information en vue d'un transfert informatique rapide par les réseaux informatiques, de restaurer une image, d'améliorer la visualisation de certains détails, de combiner des images de sources différentes ou de donner une vue tridimensionnelle d'une structure. Elles sont aussi indispensables pour préparer l'image à une quantification.

Compression

Les méthodes de compression d'images réduisent la taille informatique des images sans perte significative de l'information et optimisent ainsi l'archivage et le transfert des images par les réseaux informatiques. L'objectif est de conserver une trace des résultats dans les situations qui n'autorisent pas une conservation longue de l'échantillon original.

Restauration, amélioration et segmentation

Les sources de dégradation d'une image (flous, distorsion géométrique, hétérogénéité de l'éclairement) sont multiples et dépendent du processus de formation de l'image. La **restauration** d'images consiste à retrouver l'image la plus proche de celle qui serait obtenue avec un système d'acquisition parfait. Des procédés de déconvolution numérique sont ainsi appliqués pour optimiser la visualisation d'une image. Au sens large, l'**amélioration** d'images consiste à transformer une image pour faire ressortir une information précise contenue dans cette image. Contrairement à la restauration, elle ne cherche pas obligatoirement à approcher l'image idéale. Elle vise un objectif bien précis défini par l'opérateur, au détriment bien souvent d'autres objectifs. La **segmentation** correspond à l'ensemble des traitements qui permettent de diviser une image en plusieurs parties possédant des propriétés similaires définies par l'opérateur. C'est une phase préalable à toute quantification pour permettre à la machine d'identifier les zones ou les éléments d'intérêt.

Analyse

L'analyse d'images est la dernière étape, et nécessite que les deux premières soient résolues. Elle consiste à extraire de façon quantitative l'information contenue dans une image, et à donner une description objective, finalisée et précise de l'image, ou de certains éléments. Elle part d'une image et conduit à l'évaluation de mesures. Elle conduit à une évaluation quantitative ou à une reconnaissance de formes. Elle se pose chaque fois que l'on veut préciser une observation visuelle ou la remplacer par une analyse automatique. Elle s'est développée principalement à partir du domaine de la microscopie optique quantitative et de la robotique.

2.2. Analyse quantitative de structures

L'analyse quantitative d'images répond à différents objectifs. Elle permet de décrire une structure, de contrôler une situation par rapport à un modèle, de comparer différentes situations expérimentales ou cliniques, ou de suivre une évolution dynamique.

Analyse globale, individuelle ou locale

Une analyse est dite **globale** ou **champ par champ** par opposition à individuelle lorsqu'elle porte sur une structure dans son ensemble, quel que soit le nombre d'éléments qui la constituent. Une analyse **individuelle** est une analyse **particule par particule**, qui nécessite que l'ensemble analysé soit constitué d'éléments disjoints.

Une analyse est dite **locale** lorsque l'image ne représente qu'une partie de la structure, coupée par la fenêtre de mesure, et l'analyse doit en tenir compte. En effet, la probabilité pour un élément d'être totalement inclus dans la fenêtre de mesure dépend de sa taille. Plus une particule est grosse, plus sa probabilité d'être totalement incluse dans la fenêtre de mesure est faible.

Exemple : l'évaluation de la teneur respective des différents composants d'un minéral est une analyse globale ; l'estimation de la taille ou celle de la forme d'un ensemble de particules (billes, poudres, grains, noyaux) sont des analyses individuelles.

Contrôle de qualité

Une évaluation quantitative permet de mettre en place une évaluation automatique reproductible, objective et précise d'un échantillon, permettant notamment une comparaison avec des normes. Les applications liées au contrôle de qualité sont nombreuses

Exemple : on va chercher à contrôler la granulométrie de poudres ou de graines, l'orientation des fibres de verre dans un composite ou la composition de prélèvements de houille.

Analyse médicale

En médecine, l'analyse quantitative représente un outil performant de recherche et d'aide au diagnostic. Elle permet d'acquérir des données objectives, reproductibles et précises concernant l'importance des composants tissulaires, leur organisation tissulaire, la morphologie des cellules (taille, forme, couleur), la quantité d'acide désoxyribonucléique (ADN) nucléaire, l'intensité d'un marquage et sa répartition.

Un des objectifs poursuivis est la recherche de critères précis d'identification de groupes cliniques. On cherche à traduire quantitativement les données descriptives acquises, pour une meilleure classification des cas intermédiaires ou pour la mise en place d'un processus d'automatisation.

L'analyse d'images est aussi utilisée pour rechercher de nouvelles variables de pronostic ou de diagnostic, ou comme outil de connaissance.

Exemple : en anatomie pathologique, des programmes sont mis en place dans le cadre du dépistage des cancers du col de l'utérus, et les cellules identifiées par la machine comme suspectes sont revues pour un examen approfondi par le biologiste. En biologie de la reproduction, l'analyseur évalue la vitesse et la forme de la trajectoire des spermatozoïdes. En microbiologie, on mesure les zones d'inhibition dans les tests de susceptibilité ou on rend compte de l'adhésion bactérienne.

2.3. Reconnaissance de formes

L'analyse d'images est aussi un outil de reconnaissance de formes. L'objectif principal est alors la recherche ou le contrôle d'un événement, qui correspond à une image ou à un détail particulier. Le but peut être aussi l'identification de formes en vue d'une reconnaissance ou d'un classement automatique.

Recherche et contrôle d'un événement

La recherche de situations bien définies et d'événements rares représente une voie d'intérêt de l'analyse d'images. Dans la mesure où les objets d'intérêt sont facilement identifiables de façon quantitative, l'analyse d'images peut les repérer aisément, quel que soit leur nombre. En milieu industriel, une des applications de l'analyse d'images est le contrôle d'une situation déterminée, d'une image particulière.

Exemple : la vérification des étiquettes ou l'inspection automatique de défauts : impuretés dans des gélules pharmaceutiques ou fentes, nœuds ou poches à résine dans le bois.

Identification de formes

L'analyse d'images peut conduire à la reconnaissance automatique de formes. Nous pouvons citer comme exemple la reconnaissance automatique des caractères, imprimés ou non, en vue d'un tri automatique des lettres ou même une lecture automatique ou, en biologie, la réalisation du caryotype ou le classement de zooplanctons. On peut aussi chercher à reconnaître automatiquement des textures comme celles représentatives de types de corrosion.

3. ACQUISITION D'UNE IMAGE NUMERIQUE PAR CAMERAS CCD (CHARGE COUPLED DEVICE)

Source : Technique de l'ingénieur : traitement des images en temps réel (<http://www.techniques-ingenieur.fr>)

L'acquisition d'images à partir des caméras CCD (charge coupled device) repose sur deux principes. Le premier concerne les propriétés électro-optiques du matériau utilisé (silicium). En effet, la réponse spectrale du silicium s'étalant entre 400 nm et 1100 nm permet à ce matériau d'absorber des photons de lumière à partir du proche ultraviolet jusqu'à l'infrarouge. Le deuxième principe concerne la technologie CCD, introduite en 1970 par Boyle et Smith des laboratoires Bell. Les dispositifs CCD se présentent sous la forme d'une série de capacités MOS (metal-oxide semiconductor) couplées, constituant ainsi des registres analogiques où les charges électriques représentant l'information à transmettre sont stockées puis décalées en série vers la sortie du dispositif. Ainsi, l'intégration dans une même « puce » électronique de séries de photoéléments associés à des registres analogiques CCD permet d'obtenir des capteurs d'images à l'état solide. À l'aide de ces circuits, l'image est acquise en deux temps :

- un temps d'intégration ;
- un temps de transfert.

Durant le temps d'intégration, la lumière incidente, d'énergie $h\nu$, est absorbée par les photoéléments puis transformée en paquets de charges électriques proportionnelles à l'intensité lumineuse incidente (figure 1a). Les paquets de charge sont alors transférés vers les registres analogiques via une porte de transfert et décalés en série vers la sortie du CCD (figure 1b).

L'organisation topologique des photoéléments permet de définir deux types de capteurs d'images CCD :

- les capteurs d'images CCD linéaires (§ 5.1) ;
- les capteurs d'images CCD matriciels (§ 5.2).

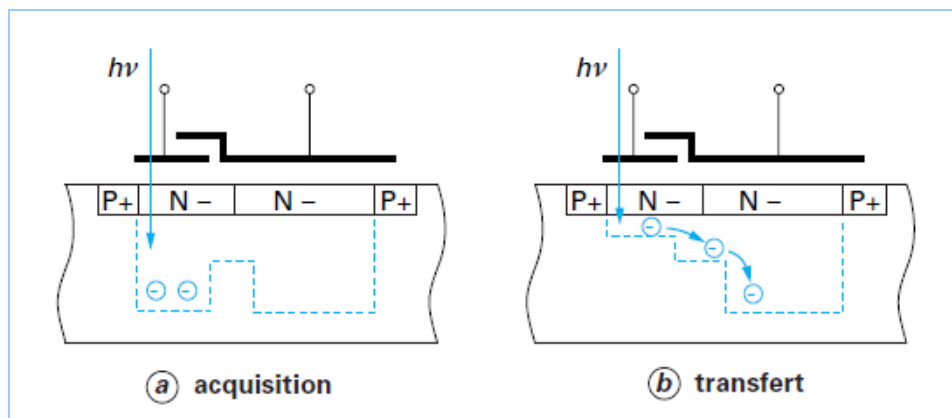


Figure 1: Principe de l'acquisition d'une image CCD

3.1. Capteurs linéaires

Ces circuits sont composés d'une rangée d'éléments photo-sensibles, d'un registre analogique (CCD) et d'un amplificateur de sortie. La figure 2 représente l'architecture d'un capteur d'image CCD linéaire. L'information lumineuse (image) est acquise et transformée en charges électriques au niveau des photoéléments (temps d'exposition). Puis dans un deuxième temps, ces charges

électriques sont envoyées vers le registre à décalage analogique (CCD) qui va ainsi transmettre ces informations en série vers la sortie vidéo sous forme de tensions analogiques échantillonnées. Il est à remarquer que pendant qu'une ligne d'image est intégrée, la ligne précédente est en train de se vider, ce qui impose une fréquence maximum f_M d'utilisation de ces capteurs selon la relation :

$$f_M = \frac{1}{N_p T_{\phi_T}}$$

Avec N_p nombre de pixels, T_{ϕ_T} période d'un pixel.

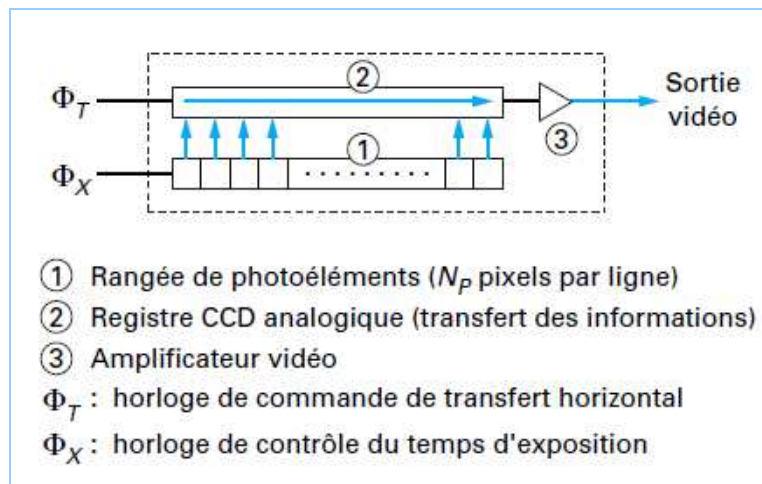


Figure 2: Capteur d'image CCD linéaire

Trois paramètres fondamentaux caractérisent ces circuits :

- le premier paramètre concerne le nombre de photoéléments (résolution) qui peut atteindre 10 000 éléments pour les plus performants ;
- le deuxième paramètre caractérise la fréquence de sortie des informations qui peut atteindre 20 MHz (horloge ϕ_T) ;
- le troisième paramètre s'intéresse à la dimension des photoéléments (de l'ordre de $10 \mu\text{m} \times 10 \mu\text{m}$) et définit ainsi la précision dans les mesures optiques.

L'image d'un objet étant à deux dimensions (lignes, colonnes), il faut donc générer un balayage mécanique du CCD ou de l'objet dans la direction perpendiculaire à la ligne photosensible afin de reconstituer la seconde dimension.

Ces capteurs d'images trouvent leurs applications dans les systèmes de reconnaissance de caractères, de mesures sans contact, de reconnaissance de formes et de détection de défauts.

3.2. Capteurs matricielles

Comme leur nom l'indique, ces capteurs sont composés de photoéléments organisés en matrice et, contrairement aux capteurs linéaires, le balayage de l'image n'est plus mécanique mais électronique. Le principe d'acquisition de l'image reste le même, toutefois deux types de configurations existent dans le concept de décalage de l'information vers la sortie vidéo :

- les capteurs matriciels à transfert de trame ;
- les capteurs matriciels à transfert par interlignes.

A. Capteurs matriciels à transfert de trame

Comme cela est montré dans la figure 3, ces capteurs sont découpés en deux parties :

- une zone photosensible, ou zone image, permettant l'acquisition de l'information lumineuse ;
- une zone mémoire permettant le stockage de cette information.

L'information lumineuse est dans un premier temps acquise dans la zone photosensible puis transférée vers la zone mémoire et décalée ligne par ligne vers le registre de lecture, si bien que lorsqu'une image est en cours d'acquisition, le contenu de la zone mémoire correspondant à l'image précédente se vide. Pour être compatible avec les systèmes vidéo, le vidage de la mémoire se fait en mode entrelacé, ce qui permet de retrouver deux trames pour une image : une trame paire et une trame impaire. Comme pour les capteurs d'images linéaires, différents signaux d'horloges sont à appliquer pour la mise en œuvre de ces circuits (ϕ_P , ϕ_M et ϕ_H).

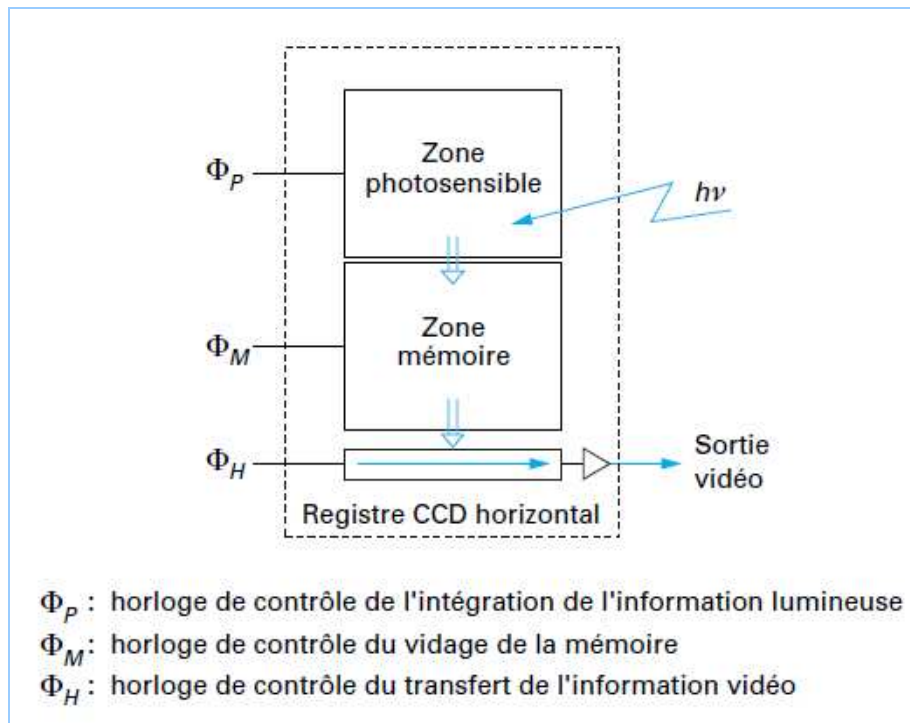


Figure 3 : Capteur CCD matriciel à transfert de trame

B. Capteurs matriciels à transfert par interlignes

Dans cette configuration (figure 4), nous retrouvons toujours la zone image et la zone mémoire, mais à chaque colonne de photoéléments correspond un registre mémoire (ou registre CCD). Ici, l'information lumineuse est acquise au niveau des colonnes de photoéléments puis transférée vers les registres CCD verticaux. Cette information est alors décalée ligne par ligne vers le registre horizontal et ressort ainsi point par point vers la sortie vidéo. Là aussi, pour être compatible avec le rythme vidéo, la découpe de l'image en deux trames se fait au niveau du transfert du contenu des photoéléments vers les registres verticaux : dans un premier temps, l'information des photoéléments de rang impair est transférée, constituant ainsi la trame impaire qui est vidée selon le processus décrit précédemment, puis dans un deuxième temps les photoéléments de rang pair sont transférés, constituant ainsi la trame paire. Pour ce type de capteur CCD, un certain nombre d'horloges de commande sont aussi nécessaires (ϕ_P , ϕ_M et ϕ_H).

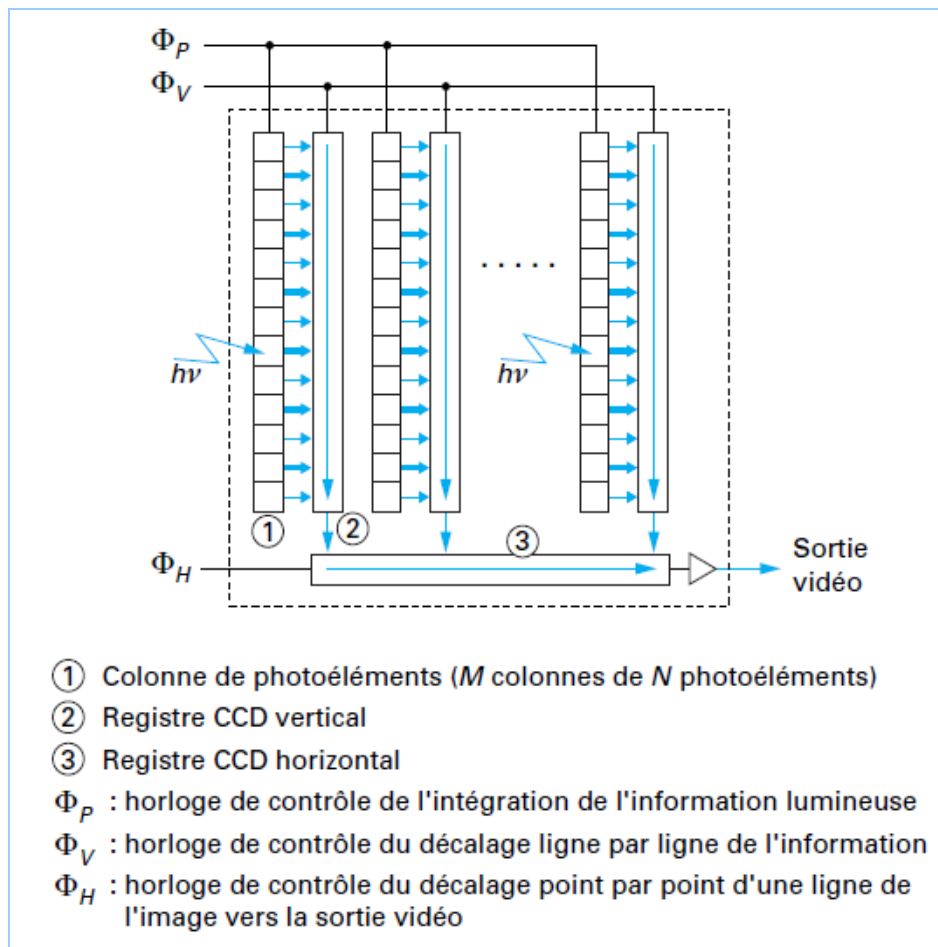


Figure 4 : Capteur CCD matriciel à transfert par interlignes

Les capteurs CCD matriciels sont à la base des caméras industrielles. Actuellement, ces caméras présentent des résolutions de 512×512 pixels jusqu'à $4\,096 \times 4\,096$ pixels selon les modèles. Les caméras CCD matricielles à transfert par interlignes sont plutôt utilisées dans les applications de vidéo instantanée ou de vidéo rapide, alors que les caméras CCD matricielles à transfert de trame, de par leurs bonnes propriétés géométriques (pixels jointifs), sont plutôt destinées à des applications de métrologie.

4. PRESENTATION DE LA PLATE-FORME DE VISION INDUSTRIELLE

4.1. Introduction

Afin de fournir aux étudiants des éléments pratiques de la vision et du traitement d'image dans le cadre d'une application industrielle, il est nécessaire de disposer d'outils performants. Ces outils ont pour rôle l'acquisition des images et le développement des traitements pour l'interprétation de ces images. Ils seront mis en œuvre dans le cadre de 2 séances de TP de 4h chacune. Ces séances doivent être indépendantes pour des raisons de mises en place de la grille des rotations de TP. Les étudiants devront ainsi pouvoir traiter les 2 TP dans un ordre quelconque

4.2. Description

TP INSPECTOR: prise en main d'un logiciel industriel d'analyse d'image (appelé *INSPECTOR*)

La démarche pédagogique sera de confronter tous les étudiants à la mise en place d'un système de vision industrielle. Cette démarche a pour but de répondre à une série de choix sur les réglages du système de vision et ses conséquences sur les traitements et l'interprétation des images.

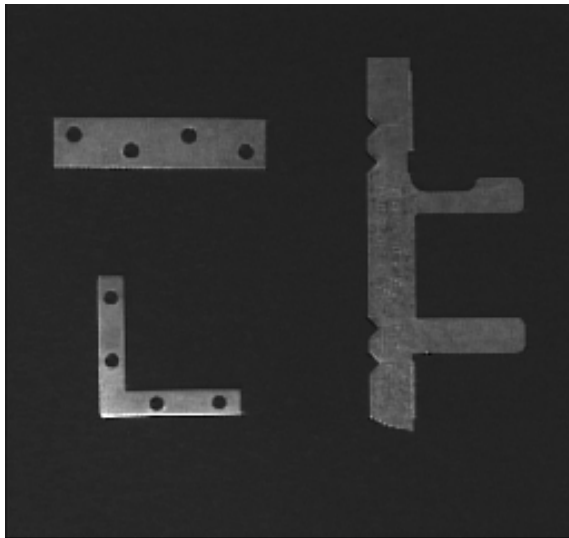
- 1) la définition du système optique (objectifs, caméra, carte d'acquisition) permet d'optimiser la résolution de la caméra à la taille de l'objet observé. Les caractéristiques de l'image numérique (quantification, échantillonnage, histogramme) seront étudiées;
- 2) la définition de l'éclairage permet de réduire la complexité des traitements (qui deviendraient nécessaires pour compenser un éclairage inadapté). On étudiera l'histogramme des niveaux de gris, le recadrage et la calibration de l'histogramme;
- 3) la définition des algorithmes de traitement d'images en fonction de la tâche de reconnaissance (contrôle qualité, métrologie, détection, identification). Le seuillage de l'image, les opérations morphologiques et le filtrage seront traités dans cette partie. Les traitements mis en œuvre seront évalués en fonction des conditions d'éclairage et en fonction de l'orientation de l'objet ;

TP MIL: prise en main d'une librairie industrielle d'analyse d'image (appelé *MIL*)

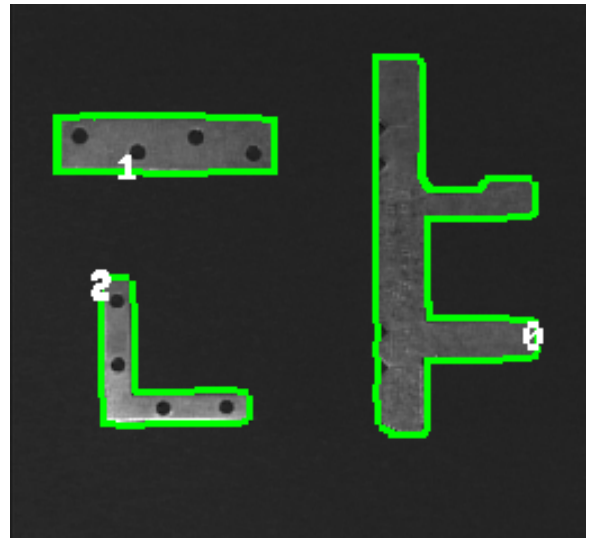
Le but de cette partie de la formation est de familiariser les étudiants à l'utilisation d'une librairie industrielle haut niveau d'analyse d'image programmée en C. Pour ce faire, les étudiants devront développer leur propre traitement d'images afin de répondre à des objectifs déterminés tels que :

- l'affichage d'une image, la lecture d'histogramme,
- le seuillage, l'ouverture et la fermeture morphologique,
- le dénombrement de pièces, l'extraction de contour,
- le calcul de périmètre d'objets détectés.

La figure ci-dessous donne un exemple de détection de contour dans le but de calculer des caractéristiques de pièces analysées.



Exemple d'image à analyser



Résultat de l'analyse

4.3. Moyens matériels et logiciels

Deux postes de TP sont mis en place. L'un est destiné à l'acquisition et à l'utilisation de traitements. L'autre est utilisé pour le développement de traitement d'analyse d'images à partir d'une librairie programmée en C. Les 2 postes sont composés comme suit.

Poste étiquette INSPECTOR :

- un PC
- une caméra CCD monochrome numérique sur support réglable
- un logiciel interactif Matrox INSPECTOR pour l'analyse d'images

Poste étiquette MIL :

- un PC
- une librairie industrielle d'analyse d'image MIL (Matrox Imaging Library)

TP INSPECTOR : ACQUISITION ET ANALYSE D'IMAGES AVEC LE LOGICIEL MATROX INSPECTOR

1. PRISE EN MAIN DU LOGICIEL INSPECTOR

Matrox Inspector est une application logicielle interactive permettant : l'acquisition d'images par caméra ou la lecture d'images sur fichiers, l'analyse et le traitement de ces images. Le paragraphe ci-après passe en revue les grandes fonctionnalités. Des définitions des traitements et des termes utilisés sont disponibles à partir de l'aide en indiquant dans la fenêtre rechercher 'glossary of terms'.

Les opérations de traitement d'images suivantes peuvent être réalisées par ce logiciel : les opérations statistiques sur l'image, le filtrage spatial, la transformée de Fourier, les opérations morphologiques, les opérations géométriques telles que : redimensionnement par interpolation ou décimation, rotation, correction de distorsions, ou encore, des opérations de reconnaissance de formes, reconnaissance de caractères, analyse de code barres, des opérations de traitement de groupes de pixels (Blob analysis), la détection de contours, la métrologie. Toutes ces opérations peuvent être appliquées à une région d'intérêt rectangulaire (ROI) ou non.

Ouvrir une session Windows avec le login et mot de passe tpvision. Brancher la caméra avant de lancer les logiciels INSPECTOR et INTELLICAM.
Les images citées dans le texte (circuit.pcx, circuit2.pcx, morpho.pcx ...) sont disponibles dans le répertoire d:\TP_vision\images_du_TP

1.1. Acquisition d'images

L'acquisition des images s'effectue avec l'application INTELLICAM. Une fois les images acquises, les sauvegarder, puis les relire avec le logiciel INSPECTOR pour réaliser les traitements demandés.

- A titre d'exemple, effectuez l'acquisition (menu Figure 1) d'une '**image A**' du circuit électrique côté composant avec la caméra du TP. Après avoir réglé la mise au point et la luminosité, figez l'image en sélectionnant le bouton . Retournez le circuit pour visualiser le côté piste du circuit et effectuez l'acquisition de l'**image B** dans ces conditions.
- Relancer une acquisition et modifiez la luminosité de la caméra pour assombrir ou éclaircir l'image. Effectuez l'acquisition de cette image (**image C**). Retournez le circuit pour visualiser le côté composant et effectuez l'acquisition de l'**image D** dans les conditions choisies précédemment de sous ou sur-expositions.

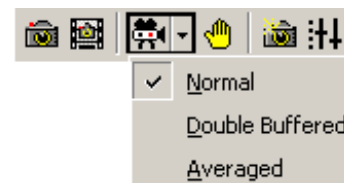


Figure 1. Menu acquisition d'images.

1.2. Histogramme

L'histogramme caractérise une image du point de vue statistique. Avant de chercher une solution à un problème de traitement d'images, il est très important de visualiser l'histogramme typique des images à traiter. Apprenez à interpréter l'histogramme et à modifier les images par transformation d'histogramme.

Exercice : si vous ne l'avez pas encore fait, lancer l'application INSPECTOR. Exercez-vous à interpréter les histogrammes des 4 images acquises . Expliquez les formes de chacun de ces histogrammes. Rappelons que l'abscisse représente les niveaux de gris, du noir (0) au blanc (255), et que l'ordonnée représente le nombre de pixels ayant ce niveau. En normalisant l'histogramme par le nombre total de pixels dans l'image, on obtient une approximation de la fonction de probabilité. En effet, si l'on prend au hasard un pixel, l'histogramme normalisé nous informe quelle est la probabilité que ce pixel ait un niveau de gris donné. Vous pouvez également sélectionner une région d'intérêt (ROI) de l'image  et visualiser dynamiquement l'histogramme en modifiant la position de cette région.

1.3. Négatif d'une image

Une des transformations d'histogramme les plus élémentaires est la visualisation du négatif d'une image (bouton  et choisir 'negate'). Appliquer cette transformation à une des images acquises et visualiser l'histogramme.

Comme toutes les transformations d'histogramme, elle peut être réalisée en faisant appel à une table de conversion (LUT), c'est-à-dire une mémoire où les niveaux d'entrée servent d'adresses, et les cases adressées contiennent les niveaux de sortie correspondants (dans le cas du négatif, à l'adresse 0 on trouve la valeur 255, à l'adresse 1 on trouve 254 etc.).

C'est la méthode la plus rapide, surtout pour des transformations plus complexes à calculer.

Exercice : pour comprendre le calcul du négatif, calculer ce négatif en utilisant la définition : $I' = 255 - I$. Pour cela, utilisez la soustraction d'images ( + 'subtract' + S1=255 + new + apply). Vérifiez que S2 () correspond à l'image à soustraire (Figure 2). Comparer les 2 images obtenues à partir du calcul du négatif et de la soustraction, en effectuant le calcul de la différence de ces 2 résultats par exemple.

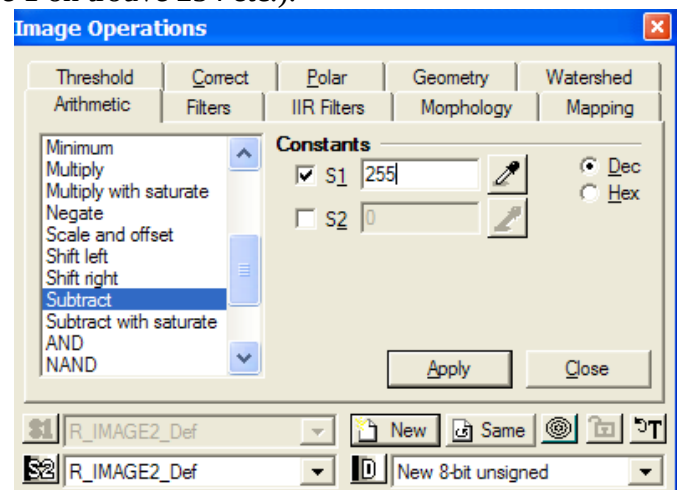


Figure 2. Soustraction d'images.

1.4. Manipulation d'histogramme

Recadrage et calibration d'histogramme

Quelle que soit l'utilisation des images acquises, il est évident que l'acquisition doit être réalisée avec un grand soin, pour assurer une bonne qualité des images. Néanmoins, il peut arriver qu'une image soit de qualité médiocre, notamment en termes de contraste. Typiquement, l'histogramme

d'une telle image est "ratatiné" à gauche (image sous-exposée), à droite (image surexposée) ou au milieu (image grise sans noir ni blanc). On peut améliorer artificiellement le contraste par transformation linéaire de l'histogramme. Le recadrage est une opération automatique : le logiciel recherche le niveau le plus sombre I_{min} présent dans l'image et le ramène à zéro, de même le niveau le plus clair I_{max} est ramené à 255. Les niveaux intermédiaires sont transformés selon la relation : $I' = a \cdot I + b = a \cdot I - a \cdot I_{min} = a \cdot (I - I_{min})$, où le coefficient $a = 255 / (I_{max} - I_{min})$ représente le "gain" en contraste.

La calibration (Figure 3) réalise exactement la même relation, mais les niveaux I_{min} , I_{max} ne sont pas déterminés automatiquement et peuvent être différents des niveaux minimum et maximum présents dans l'image. Il en résulte alors une saturation en blanc des niveaux supérieurs à I_{max} et l'effacement des niveaux inférieurs à I_{min} . En imagerie médicale par exemple, le médecin radiologue effectue cette opération très fréquemment, pour mettre en exergue des détails situés dans une plage de niveaux de gris particulière. Il sacrifie alors volontairement les autres détails.

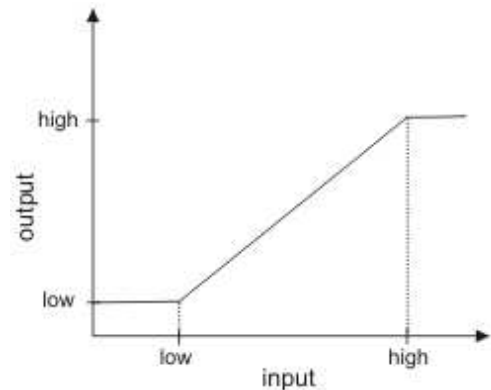


Figure 3. Calibration d'histogramme (Window Leveling)

Exercice : dans cette partie, il vous est proposé de travailler sur les images côté composants : les images **A** et **D**. Essayez, par exemple, à partir de l'**image D**, de trouver la calibration qui donne le meilleur contraste en vous rapprochant de l'**image A**. Affichez l'histogramme de l'image A, effectuez une calibration d'histogramme (bouton  + WL=Window Leveling + output range min et max) pour que l'image résultante se rapproche de l'image A.

Ajustement linéaire du contraste et de la luminosité

Le principe est décrit sur la Figure 4.

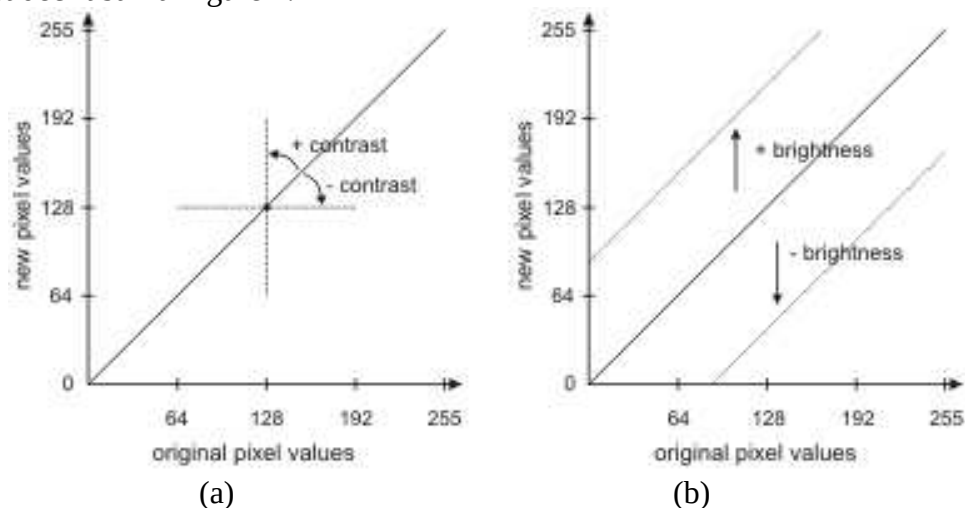


Figure 4. Ajustement linéaire du contraste (a) et de la luminosité (b).

Exercice : Afficher l'histogramme de l'image D. Ajuster le contraste puis la luminosité de l'image (bouton  + BC=Brightness/contrast + valeur à ajuster) en prenant des valeurs extrêmes et intermédiaires pour bien comprendre le principe.

Correction Gamma

La correction Gamma a pour effet de modifier l'intensité d'un pixel selon l'équation :

$$new = maxvalue \times \left(\frac{value}{maxvalue} \right)^{\frac{1}{\gamma}} \quad (1.1)$$

Il est évident que pour $\gamma = 1$, l'intensité du pixel est inchangée. Cela correspond à la droite représentée sur la Figure 5.

Exercice : Afficher l'histogramme de l'image D. Appliquez la correction Gamma (bouton  + GC=Gamma Correction + valeur à ajuster) à l'image en prenant des valeurs extrêmes et intermédiaires pour bien comprendre le principe.

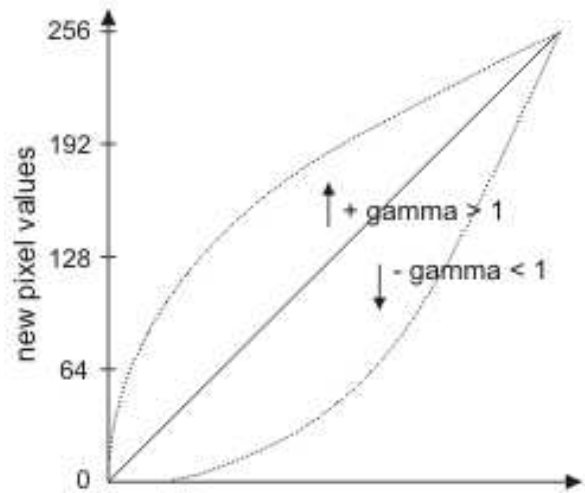


Figure 5. Correction Gamma.

Egalisation d'histogramme

Dans une image naturelle qui a été quantifiée de manière linéaire, une majorité de pixels ont une valeur inférieure à la luminance moyenne. C'est pourquoi les détails dans les régions sombres sont difficilement perceptibles. Pour palier cet inconvénient, on peut recourir aux techniques de transformation d'histogramme, parmi lesquelles, on peut citer les transformations de forme exponentielle, hyperbolique ou uniforme. Cette dernière est connue sous le nom d'égalisation d'histogramme.

	Output probability density model	Transfer functions
Uniform	$p_g(g) = \frac{1}{g_{\max} - g_{\min}}$ $g_{\min} \leq g \leq g_{\max}$	$g = [g_{\max} - g_{\min}] p_f(f) + g_{\min}$

$$p_f(f) \approx \sum_{m=0}^i H_f(m)$$

La fonction de répartition $p_f(f)$ est approchée par l'histogramme cumulé :

Exercice : 1) Afficher l'histogramme de l'image D. 2) Appliquer l'égalisation d'histogramme (bouton  + HE1=égalisation uniforme d'histogramme + valeur à ajuster) à l'image. 3) Afficher l'histogramme de l'image résultante. Effectuer une deuxième fois les opérations de 1) à 3) et comparer les résultats obtenus.

Seuillage

Le seuillage peut être vu comme une transformation d'histogramme. Le seuillage simple consiste à répartir les pixels d'une image en deux classes selon leur niveau de gris comparé à un seuil. On peut afficher les numéros des classes comme niveaux de gris : noir pour les pixels dont le niveau initial était inférieur ou égal au seuil, et blanc pour les autres pixels. On dit alors qu'on a obtenu une image binaire (deux niveaux). Le seuil peut être choisi manuellement ou fixé automatiquement en s'appuyant sur des hypothèses concernant l'histogramme de l'image.

Exercice : Afficher l'histogramme de l'image B. Appliquer l'opération de seuillage (bouton  + seuillage simple + valeur à ajuster) à l'image. Afficher l'histogramme de l'image C et appliquez l'opération de seuillage sur cette image. Comparer les deux résultats obtenus.

1.5. Quantification

Pour simplifier la manipulation de la mémoire, les images vidéo sont généralement numérisées (quantifiées) sur 1 octet (8 bits) par pixel, ce qui donne 256 niveaux de gris. Mais l'œil humain distingue seulement une trentaine de niveaux de gris.

- 1- Charger une image du circuit.
- 2- Réduire le nombre de bits utilisés pour la quantification. Pour cela, vous effectuerez des décalages de valeurs différentes ( + 'shift right' + valeur), puis recadrer l'image résultat. En particulier, interpréter vos résultats pour des décalages de 7 bits à droite, 5 bits et 1 bit.
- 3- Conclusions ?

La Figure 6 vous donne une indication sur les résultats à obtenir et à commenter.

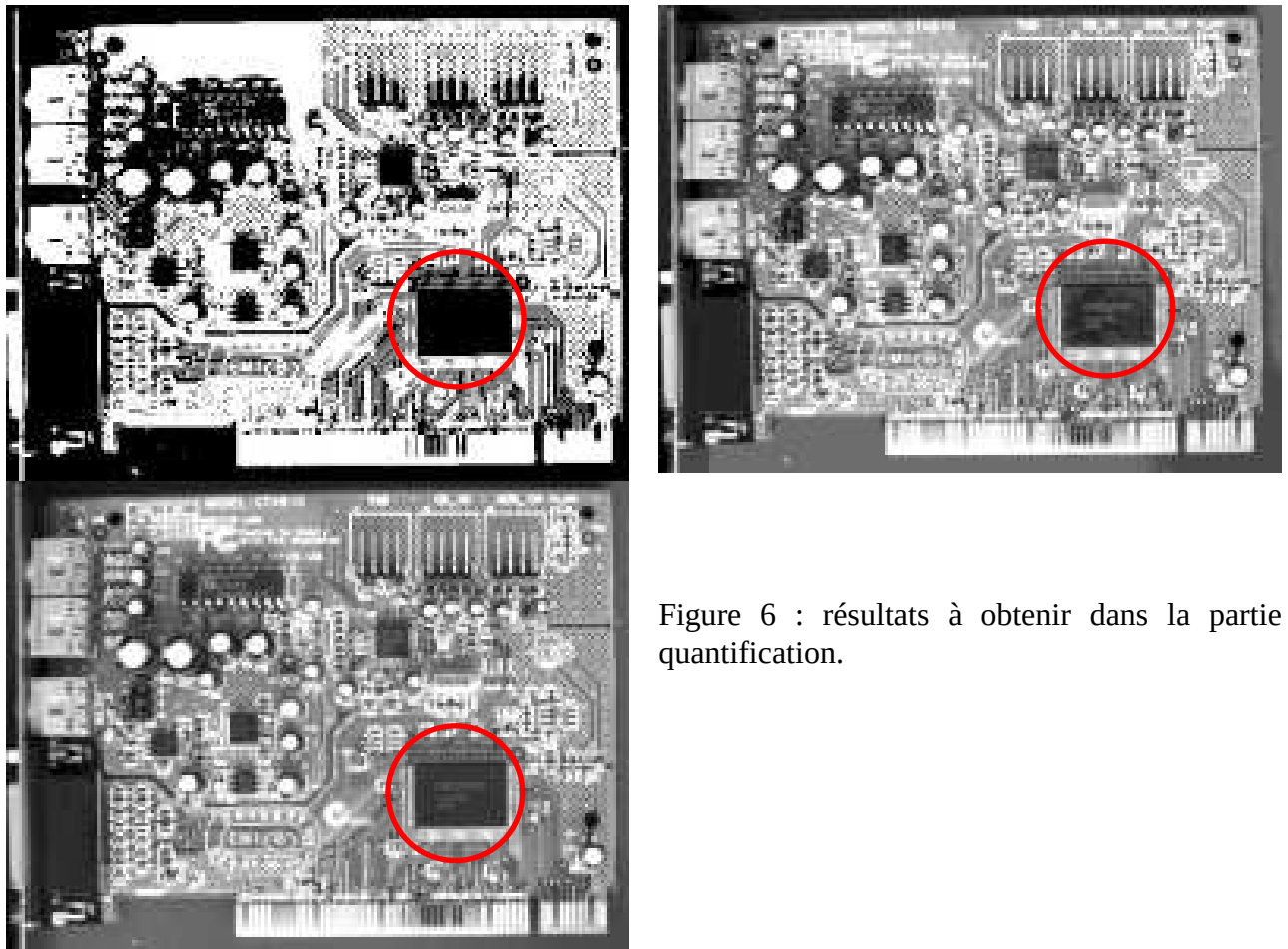


Figure 6 : résultats à obtenir dans la partie quantification.

1.6. Opérations morphologiques (voir annexes)

La morphologie mathématique fournit un ensemble d'opérateurs destinés à l'analyse de formes présentes dans l'image. Ces définitions sont données pour les images binaires, mais la plupart des opérations peut également être définie pour les images à niveaux de gris. Retenez l'importance de la forme de l'élément structurant (outil) utilisé.

Erosion et dilatation

Ces deux opérations, érosion et dilatation, sont complémentaires. Vous verrez qu'elles permettent "d'usiner" les bords des objets clairs et sombres respectivement. Elles permettent aussi de définir tous les autres opérateurs morphologiques.

Exercice : utiliser l'image 'morpho.pcx' ou sinon les images acquises côté pistes (images B et C) . Faire un seuillage de l'image. On veut éliminer les objets de petite dimension. Pour ce faire, on va réduire la taille de tous les objets jusqu'à disparition des petits objets, puis on va augmenter la taille des objets jusqu'à leur taille initiale.

1- Définir l'élément structurant : choisir par exemple l'élément structurant suivant ( + **New...** + **Pre-defined...** + H 3x3 + OK)

2- Eroder l'image avec un nombre d'itération imposé ( + erode + apply to = dark /bright objects suivant l'image + iterations=12)

3- Dilater (dilate + iterations=12) avec le même élément structurant.

4- Partant de l'image obtenue à l'étape 3, effectuer une dilatation en choisissant un élément structurant en croix à 4 voisins que vous personnaliserez comme sur la Figure 7. Appliquer 5 fois cet élément pour dilater l'image de l'étape 3.

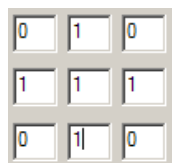


Figure 7. Masque ou élément structurant en croix à 4 voisins à définir comme suit :  + **New...** + masque défini + OK.

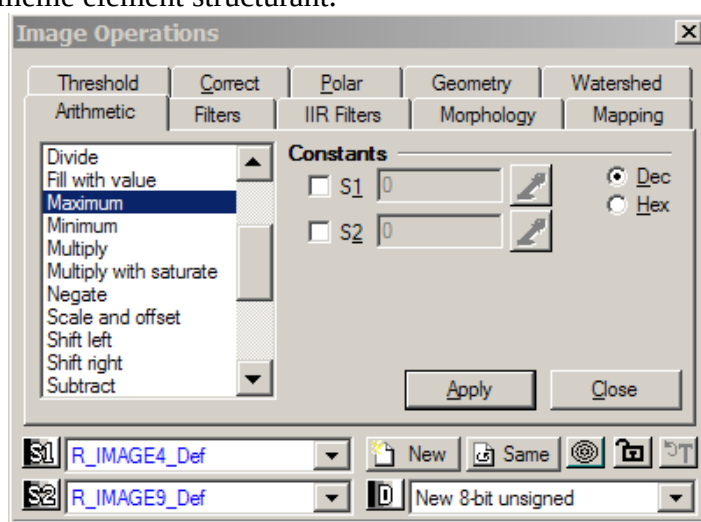


Figure 8. Exemple d'utilisation du maximum entre les images 'R_IMAGE4_Def' et 'R_IMAGE9_Def'.

5- Calculer le maximum entre cette image et l'image originale après seuillage en vous aidant de la Figure 8.

6- Refaire les opérations 1 à 5, en travaillant directement avec l'image originale (sans seuillage). Conclusions ?

Ouverture et fermeture

Pour diverses raisons (bruit par exemple), les formes présentes dans une image sont souvent "rugueuses". En combinant l'érosion et la dilatation, on obtient les opérateurs d'ouverture et de fermeture qui enlèvent les aspérités et les petites taches. Par définition, l'ouverture consiste à faire une érosion, utilisant un élément structurant ou masque donné, suivie d'une dilatation avec le masque symétrique par rapport au précédent. Lorsque le masque de départ est déjà symétrique (c'est notre cas), les deux opérations se font avec le même masque.

Exercice : utiliser les images acquises côté pistes (images B et C) ou sinon l'image 'circuit2.pcx'.

- 1- Réaliser l'opération de fermeture, avec un masque carré de taille 5x5. Le masque 5x5 s'obtient en itérant 2 fois le masque 3x3. Quelles sont les parties qui ont été modifiées ?
- 2- Faire un seuillage de l'image (par exemple régler le seuil à 60 pour l'image 'circuit2.pcx').
- 3- Réaliser l'ouverture de cette image avec le masque carré de taille 5x5. Quel est l'effet obtenu ?
- 4- Réaliser la fermeture de l'image binaire avec le masque carré de taille 5x5. Quel est l'effet obtenu ?

Squelettisation

La squelettisation d'une image binaire extrait le squelette ou encore les "lignes de force" d'une image. Elle s'effectue par une succession d'opérations appelées amincissements jusqu'à obtention d'une structure stable où les lignes sont d'épaisseur 1 pixel.

Exercice : utiliser l'image 'circuit2.pcx'

- 1 - calculer le négatif de cette image
- 2- Effectuer le seuillage du négatif de l'image
- 3 - Eliminer les points blancs parasites sur l'image binaire (ouverture avec un masque en croix et une itération)
- 4 - Extraire le squelette de l'image binaire (Thin +  To skeleton). Expliquer le résultat.

Marquage

L'opérateur de marquage, défini à partir de la dilatation, est un outil précieux permettant de sélectionner, dans une image, les formes ayant une intersection non nulle avec un objet, dans une autre image, utilisé comme marque (ou germe).

Exercice :

- 1- Charger l'image morpho.pcx.
- 2- Prendre le négatif. Effectuer un seuillage pour obtenir l'image binaire des différents objets.
- 3- Effectuer des érosions pour ne garder que des pixels appartenant au plus gros objet circulaire.
- 4- Marquer l'image obtenue en 2, avec l'image obtenue en 3 ( + Reconstruct + From Seed in S2). Quel est le résultat ?
- 5- Complément : charger l'image objects.pcx. Avec cette image, réaliser les opérations nécessaires pour ne garder que les deux stylos, avec leurs niveaux de gris (le reste de l'image étant noir).

Etiquetage

L'étiquetage consiste à rechercher les ensembles de pixels connexes, ayant le même niveau (blanc ou noir). A chaque ensemble (région) ainsi formé, on affecte une étiquette = numéro de région. Attention ! En présence d'objets fins, le résultat peut dépendre de la connexité (masque) choisie !

Exercice : Charger l'image circuit2.pcx.

- 1 - Calculer le négatif de cette image.
- 2- Effectuer le seuillage du négatif de l'image (à 172 environ).
- 3 - Faire une fermeture pour remplir les pastilles
- 4 - Effectuer un amincissement puis une érosion pour éliminer les pistes, suivie d'une dilatation pour retrouver la taille initiale des pastilles.
- 5 - appliquer l'opération d'étiquetage pour différencier les pastilles (Label +  Pseudo). Expliquer votre résultat?
- 6 - Compter les pastilles ( + Extremes + LastData).
- 7 - Conserver le résultat de l'étape 4 pour la suite.

Image de distance

Il s'agit d'une transformation qui attribue à chaque pixel la distance de ce pixel à l'objet blanc le plus proche. Une application simple consiste à sélectionner les objets selon le critère de leur taille, ce qui permet, par exemple, de vérifier les cotes d'objets usinés (diamètre d'un trou...). Mais on utilise l'image de distance dans des applications plus complexes, par exemple en segmentation par contours actifs. Un contour actif est un modèle déformable qui possède certaines propriétés (continuité, forme globale prédéfinie...) et qui évolue sous l'influence de "forces" définies à partir de l'image, afin d'épouser au mieux une des formes présentes dans l'image. Une de ces forces peut être proportionnelle à la distance : elle attire alors le contour actif vers le bord d'une région et s'annule lorsque le contour atteint ce bord.

Exercice : utiliser une des images acquises ou l'image 'objets.pcx'

- 1- Effectuer le seuillage de l'image pour séparer les objets du fond.
- 2 - Calculer l'image de distance au moyen de l'algorithme city block ( + distance + city block).
- 3 - Pour mieux vous rendre compte de l'aspect de l'image de distance, faites un recadrage d'histogramme de cette image. Expliquer comment fonctionne cet algorithme en vous aidant de l'aide.
- 4 - Revenez à l'image binaire obtenue en 1. Quelles opérations permettent de sélectionner tous les objets d'épaisseur au moins égale à 20 pixels ?
- 5 - Vous pouvez changer d'algorithme de calcul de distance et visualiser les modifications sur l'image distance et sur l'image finale obtenue à l'étape 4.

Détection des trous

On conçoit aisément qu'il y a un nombre important d'applications industrielles où il est important de détecter des trous, soit pour vérifier leur présence et leurs dimensions, lorsque leur présence est désirée, soit pour détecter des anomalies.

Exercice :

- 1- Effectuer un seuillage de l'image objets.pcx, puis réaliser une fermeture, afin de supprimer les régions de petites surfaces qui ne correspondent pas à des trous (ex. points sur les dés).
- 2- Appliquer la fonction de sélection de trous ( + Reconstruct + Extract holes).
- 3 - Variante : sans utiliser la fonction de sélection des trous, effectuer les opérations morphologiques nécessaires pour obtenir le même résultat. Prendre le négatif de l'image objets.pcx. Effectuer un seuillage de cette image. Faire une ouverture (x1). Eliminer les objets du bord ( + Reconstruct +  Erase border).
- 4- Comment sélectionner automatiquement les objets troués ?

Génération d'une image de bord

Dans cette image de bord, tous les pixels sont noirs sauf les pixels du bord qui sont blancs.

- 1 - Générer une image uniforme où tous les pixels sont blancs ( + Fill with value + 255).
- 2 - Translater 2 fois cette image avec les coordonnées (x,y) = (1,1) puis (x,y) = (-1,-1) en suivant le menu ( + Translate + X = , Y =).
- 3 - Faire un ET entre les deux images obtenues à l'étape 2 ( + AND + choisir  et )..
- 4 - Faire la différence entre l'image de l'étape 1 et celle de l'étape 4 (Substract + choisir  et ) pour obtenir l'image de bord souhaitée. Quelle est l'image obtenue (faire un zoom pour vérifier les valeurs de bord)?
- 5 - Conserver cette image pour la suite.

Points terminaux

Exercice : utiliser l'image circuit2.pcx.

- 1 - Calculer le négatif de cette image.

- 2- Effectuer le seuillage du négatif de l'image (à 172 environ).
- 3 - Faire une fermeture pour remplir les pastilles
- 4 - Extraire le squelette de l'image binaire (Thin +  To skeleton)
- 5 - Récupérer les points terminaux à partir de l'image de l'étape 4 (ConnectMap +  End points)
- 6 - On souhaite éliminer les points terminaux touchant le bord de l'image pour ne conserver que les points terminaux correspondant à la position des pastilles. Proposer une solution en vous appuyant sur l'image de bord de l'exercice précédent.

Exercice de synthèse :

Utiliser l'image Des.pcx. Réaliser un enchaînement d'opérations permettant de vérifier le nombre de dés et d'effectuer un comptage automatique du nombre total des points sur les dés. A vous de jouer !

Exercice de synthèse : utilisez l'image 'circuit.pcx'. Par un enchaînement d'opérations, proposer un moyen de détecter et de visualiser la coupure de piste.

2. CHAÎNE DE TRAITEMENT

On propose de mettre en place une chaîne de traitement automatisée permettant de dénombrer le nombre de dés, puis le nombre de points total indiqué par les trois dés.

2.1. Acquérir les images

Acquérir les images des trois dés.

2.2. Les traiter par la chaîne mise en place

Faire l'acquisition de l'image des dés. Réaliser un enchaînement d'opérations permettant de vérifier le nombre de dés et d'effectuer un comptage automatique du nombre total des points sur les dés. A vous de jouer !

- 1 - Après avoir acquis l'image des dés.
- 2 - Sélectionner la commande "enregistrement de script" ().
- 3 - Donner un nom à la fonction, par exemple, "compte", assigner un raccourci Ctrl Shift C, puis valider (bouton OK).
- 4 - Effectuer l'enchaînement des traitements nécessaires pour compter les dés puis, compter les points sur les dés.
- 5 - Arrêter l'enregistrement du script à l'aide de la commande STOP ().
- 6 - Tester votre script.

TP MIL : ANALYSE D'IMAGES AVEC LA LIBRAIRIE MIL (MATROX IMAGING LIBRARY)

Matrox Imaging Library (MIL) est une librairie industrielle de haut niveau d'analyse d'images programmée en langage C. Elle a été spécialement créée pour la capture d'images, le traitement d'images, la reconnaissance de forme, l'enregistrement, l'analyse d'objet, l'extraction et l'analyse de contours, les mesures, la reconnaissance de caractères, la lecture de code 1D et 2D, la calibration, les graphiques, la compression d'images, l'affichage d'image et l'archivage

1. INTRODUCTION A LA LIBRAIRIE MIL

Avant de découvrir comment il est possible d'utiliser une telle librairie, nous allons étudier comment il est possible sous un environnement particulier (ici Microsoft Visual C++) d'inclure une librairie à un projet afin de l'utiliser. La démarche proposée devra être reproduite à chaque fois que vous devrez créer un nouveau projet qui fait appel à une librairie.

Ouvrir une session Windows avec le login et mot de passe TPVISION.

1.1. Les différentes étapes afin d'inclure une librairie dans un projet

Lorsque l'on souhaite inclure une librairie dans un projet en C ou en C++, il est nécessaire dans les propriétés du projet de spécifier les chemins d'accès aux fichiers sources (".h") et aux fichiers de librairie (".lib") correspondant à la librairie à inclure. Ainsi, nous allons suivre une procédure "efficace" qui permet, sous Visual Studio, d'effectuer une telle étape. Il est à noter que Visual Studio est un IDE (Environnement de Développement Intégré) particulier mais que la procédure présentée est générale et s'applique à tout autre IDE (tel que KDevelop, Code ::Block, Eclipse, Dev-C++, ...)

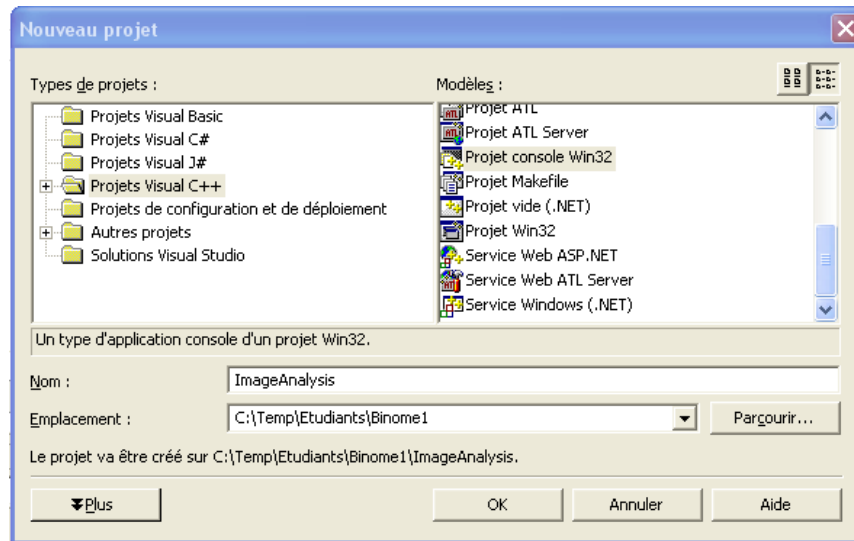
1.1.1. Création d'un espace de travail (Workspace) sous Visual Studio

A partir du raccourci sur le bureau, lancer l'application "Microsoft Visual Studio .NET 2003". Afin de créer un nouveau projet, vous devez suivre la procédure suivante :

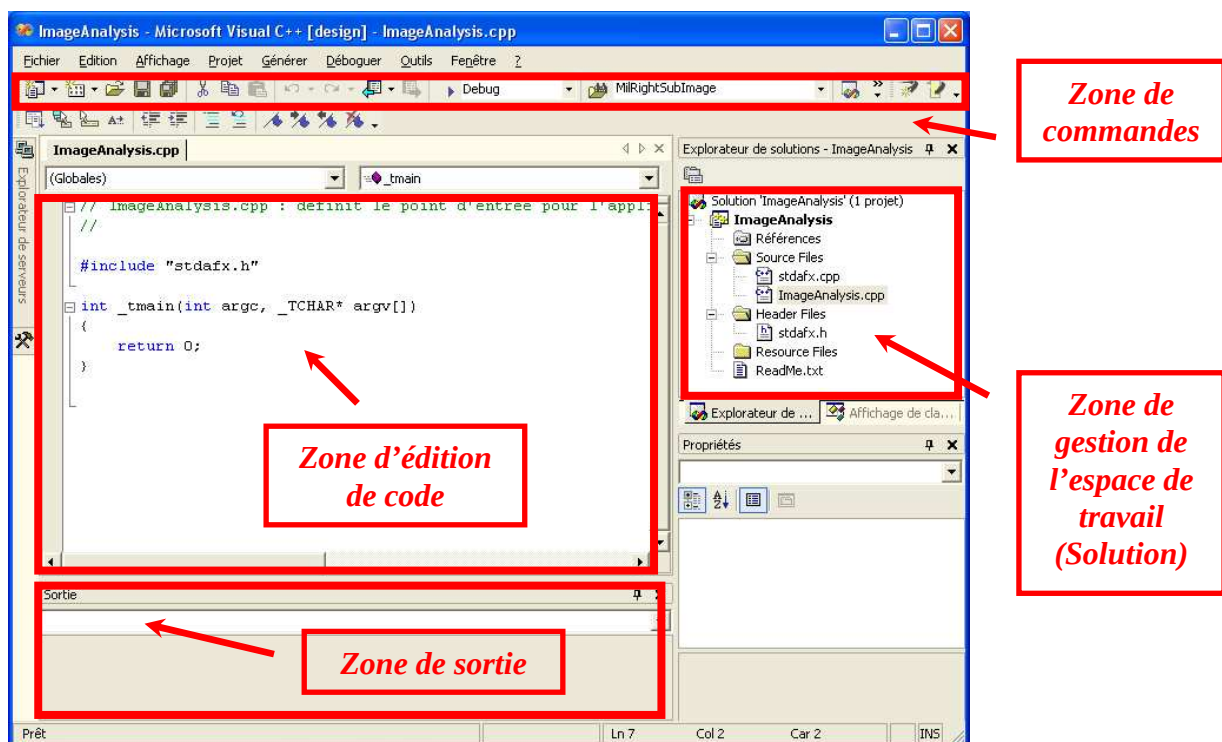
- Fichier -> Nouveau -> Projet

Vous devez ensuite choisir le type de projet. Vous pouvez voir qu'il est possible de créer un nombre important de différents types de projet sous cet IDE, tel que la création de librairie statique (.lib) ou dynamique (.dll). Dans ce tutoriel, nous allons créer un projet "Application Console" qui permet, au cours de l'exécution du programme, d'interagir avec l'utilisateur via une console. Pour ce faire, vous devez suivre la procédure suivante :

- Projets Visual C++ -> Projet console Win32 ;
- Sélectionner l'emplacement "C:\Temp\Etudiants\VotreBinome" ;
- Saisir un nom de projet, par exemple *ImageAnalysis*.



Une fois le projet créé, vous devez avoir l'espace de travail (appelé Solution) suivant



- la zone d'édition de code vous servira à coder votre application.
- la zone de commande vous permet de compiler votre projet soit en version "debug", soit en version "release". De plus vous avez accès à un certain nombre de commandes utiles lors de la compilation/exécution de votre code (par exemple pour l'exécution en mode "debugger").
- la zone gestion de l'espace de travail, aussi appelé Solution, permet de gérer et de configurer chaque projet attaché à votre Solution. Noter que lorsque vous créez un

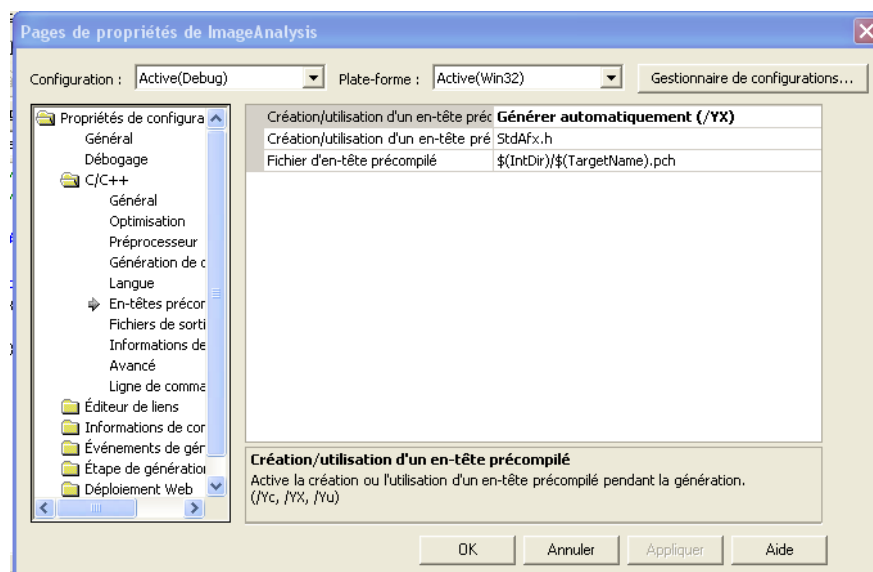
projet, Visual Studio crée une Solution et rattache un projet de même nom, ici *ImageAnalysis* ;

- La zone de sortie fournit des informations d'erreurs ou d'avertissements lors de la compilation d'un projet.

Remarque : Vous devez en 5^{ème} année GE connaître les définitions/propriétés/différences entre le mode "debug" et le mode "release" et être capable d'exécuter un code en mode "debogueur" ou "pas à pas".

Lorsque l'on analyse la zone de gestion de l'espace de travail, vous pouvez remarquer que "Visual Studio", par défaut, inclus les fichiers "stdafx.h" et "stdafx.cpp". Ce sont des fichiers de configuration optionnels que nous allons supprimer. Il suffit pour cela d'exécuter les deux étapes suivantes :

- dans la zone de gestion de travail, sélectionner chacun des deux fichiers et appuyer sur la touche "suppr" du clavier (ou faire un clic droit puis Supprimer);
- toujours dans la zone de gestion de travail, cliquer avec le bouton gauche de la souris sur le nom de votre projet (ici *ImageAnalysis*) puis cliquer avec le bouton droit et sélectionner "Propriété". Sélectionner l'onglet "C/C++" -> "En-têtes précompilé". Dans la fenêtre de droite, à la ligne "Création/utilisation d'un en-tête précompilé", sélectionner l'option "Générer automatiquement (/YX)". Enfin cliquer sur OK



Une fois que vous avez réalisé toutes ces étapes, il est indispensable de compiler votre projet afin de voir si sa configuration actuelle est bonne. Pour cela, il suffit de suivre la procédure suivante :

- Générer -> Générer la solution

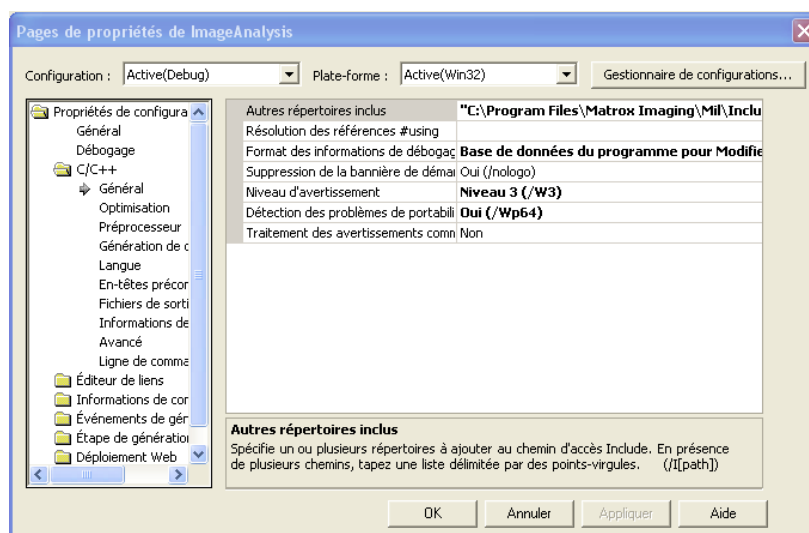
Si votre projet est configuré correctement, vous devez avoir aucune erreur et aucun avertissement dans la zone de sortie.

1.1.2. Insertion de la librairie MIL sous Visual Studio

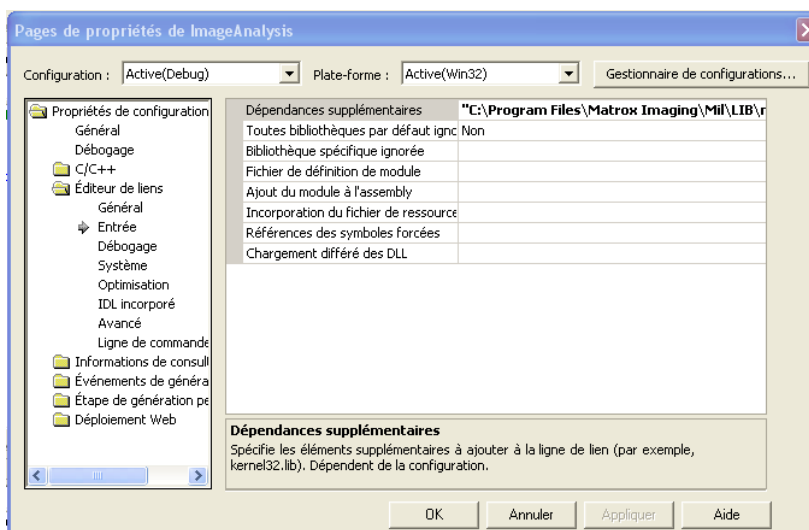
Afin d'insérer la librairie MIL dans votre espace de travail, vous devez indiquer les chemins d'accès aux fichiers sources de la librairie dans les paramètres de configuration de chaque projet de la Solution. Ces chemins seront utilisés lors de la compilation pour attacher des fonctions prédéfinies de la librairie MIL à votre projet. La procédure à suivre est la suivante :

- dans la zone de gestion de travail, cliquer avec le bouton gauche de la souris sur le nom de votre projet (ici *ImageAnalysis*) puis cliquer avec le bouton droit et sélectionner "Propriété".
- sélectionner l'onglet "C/C++" -> "Général". Dans la fenêtre de droite, à la ligne "Autres répertoire inclus", taper (ou sélectionner)
 - o "C:\Program Files\Matrox Imaging\Mil\Include"

Remarque: attention les guillemets font partie de la saisie.



- sélectionner l'onglet "Editeur de liens" -> "Entrée". Dans la fenêtre de droite, à la ligne "Dépendances supplémentaires", taper (ou sélectionner)
 - o "C:\Program Files\Matrox Imaging\Mil\LIB\mil.lib"
- "C:\Program Files\Matrox Imaging\Mil\LIB\Milim.lib"



- Cliquer sur OK

Remarque: nous avons lié les deux fichiers "mil.lib" et "Milim.lib" au projet "ImageAnalysis". Ces fichiers possèdent les fonctions de bases que nous allons utiliser lors de ce TP. Si vous devez insérer un autre fichier ".lib" à votre projet, vous devez ajouter le chemin de ce fichier au même endroit que pour les fichiers "mil.lib" et "Milim.lib".

Vous venez de suivre toutes les étapes indispensables pour pouvoir utiliser la librairie MIL au sein de votre projet. Nous allons voir maintenant comment nous pouvons utiliser des fonctions prédéfinies pour réaliser des traitements/analyses d'images de façon automatique.

1.2. Mode d'emploi d'apprentissage de la librairie MIL

Avant d'utiliser des fonctions prédéfinies de la librairie MIL, il est nécessaire d'avoir une vue d'ensemble des fonctions présentes dans cette bibliothèque. Pour ce faire, lancer l'application "Mil Control Center" dont le raccourci est présent sur le bureau. Cette application donne accès à un ensemble d'informations sur les différents outils Matrox installés sur le PC, dont la librairie MIL. Dans la rubrique MIL, vous avez accès à 4 parties :

- **Example Launcher :** permet de tester en temps réel des exemples d'analyse d'images basés sur la librairie MIL et d'étudier le code associé ;
- **Help :** permet d'avoir accès à l'outil d'aide de la librairie MIL. L'aide est un outil précieux que vous devez utiliser sans arrêt afin d'avoir des informations sur la fonction que vous souhaitez utiliser ;
- **Directory :** ouvre un explorateur Windows au niveau du dossier qui contient tous les fichiers relatif à la librairie MIL ;
- **Examples directory :** ouvre un explorateur Windows au niveau du dossier qui contient tous les fichiers d'exemple d'utilisation de fonctions de la librairie MIL.

Afin de progresser le plus rapidement possible, nous vous conseillons de procéder de la façon suivante. Tout d'abord, à partir du lien "Example Launcher", parcourez l'ensemble des exemples afin d'évaluer les possibilités d'analyse d'images de la librairie MIL. Pour ce faire, sélectionner un exemple particulier (par exemple mimprocessing – étape 1 dans la figure 1), liser dans la fenêtre de droite les informations relatives à l'exemple (étape 2 de la figure 1) puis lancer l'application en appuyant sur le bouton "Play" (étape 3 de la figure 1).

Lors de l'exécution du programme, vous pouvez interagir avec l'application en utilisant la fenêtre de commande. Une fois l'exemple terminé, vous pouvez lire le code source de l'application en cliquant sur le bouton « Open Example Directory » et en ouvrant le fichier source (.cpp) correspondant (dans notre exemple « mimprocessing.cpp » - étape 4 de la figure 1). Vous devez analyser le code, le comprendre dans son ensemble pour pouvoir le réutiliser dans votre propre code.

Afin d'analyser le code, vous allez devoir comprendre le fonctionnement de chaque fonction MIL appelée. Pour cela, vous devez utiliser l'outil d'aide MIL accessible à partir de l'application « Mil Control Center » présent sur le bureau. **Cette outil d'aide est indispensable pour pouvoir comprendre et analyser les fonctions de la librairie MIL.** Vous devez sans arrêt y faire référence au cours du TP afin de compléter votre propre code source.

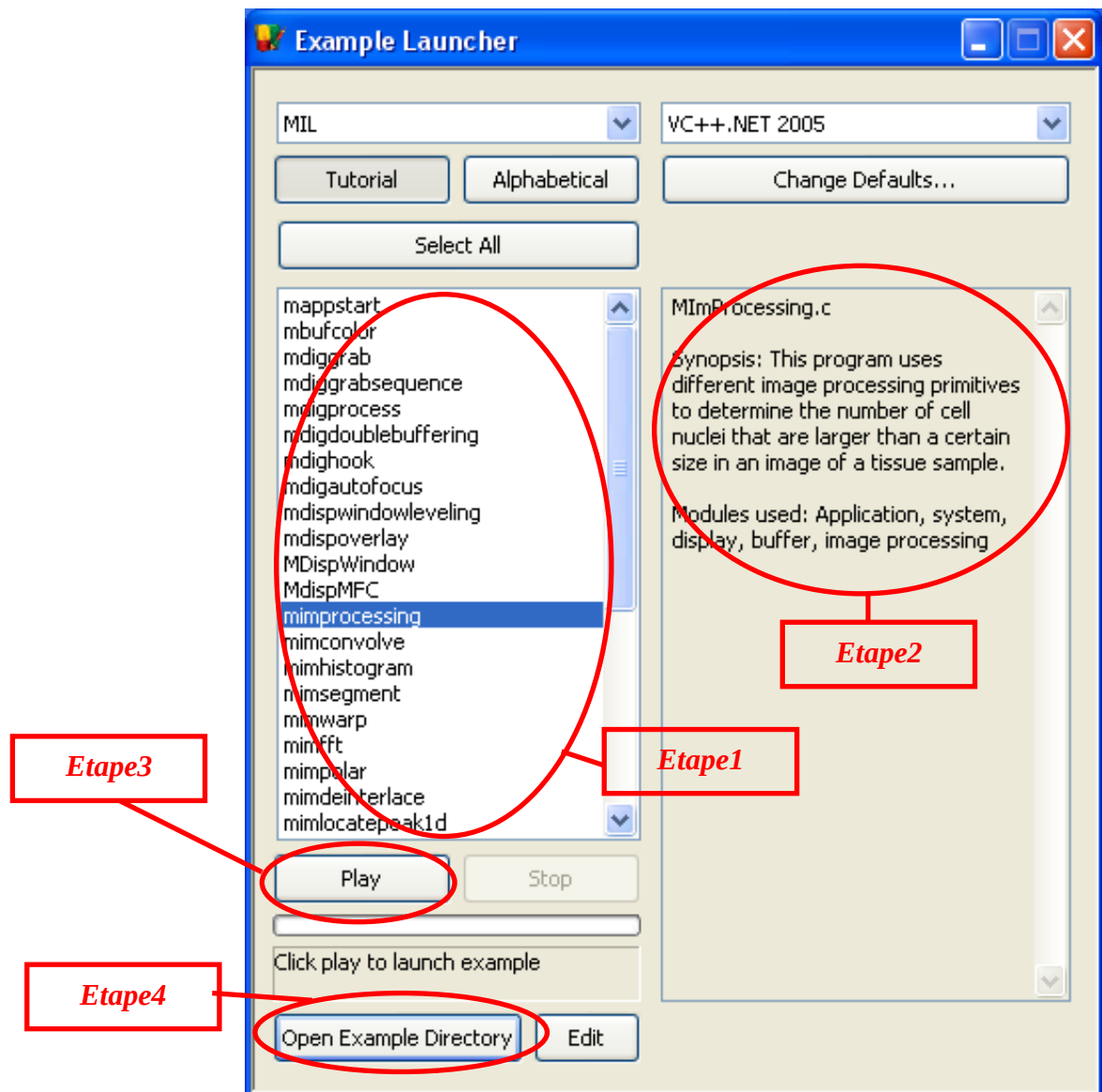


Figure 1 : fenêtre de navigation d'exemples de manipulation de fonctions définies dans la librairie MIL

2. LECTURE D'UNE IMAGE A PARTIR DE MIL

Afin de montrer la facilité de programmation grâce à l'utilisation de la librairie MIL, nous avons programmé une application qui permet de:

- lire une image dont le chemin est spécifié dans le code ;
- enregistrer les informations relatives à l'image (taille, nombre de bits utilisé pour coder l'intensité de chaque pixel) ;
- afficher l'image correspondante ;
- afficher dans la console les informations image.

2.1. Récupérer un code déjà existant

Afin d'insérer le code présenté ci-dessus dans votre projet, vous devez suivre les étapes suivantes :

- supprimer le code déjà existant de votre projet, présent dans le fichier "ImageAnalysis.cpp";
- à partir du raccourci "sources" présent sur le bureau, ouvrir le fichier texte "ImageAnalysis.txt" ;
- copier l'ensemble du texte et le coller dans votre projet, dans le fichier "ImageAnalysis.cpp".
- sauvegarder l'ensemble de votre projet (menu : Fichier -> Enregistrer tout) ;
- compiler votre projet (menu : Générer -> Générer la solution) ;
- après la compilation, vous ne devez avoir ni d'erreur, ni d'avertissement dans la zone de sortie (raccourci Ctl+Maj+B);
- exécuter votre application (menu : Déboguer -> Démarrer ou l'onglet flèche bleu ou la touche raccourci F5).

2.2. Analyse du code proposé

Afin de vous aider à analyser le code, vous devez répondre aux questions suivantes :

1. Où est déclaré le mot clé MIL_ID ? A quel type correspond-il ?
2. En utilisant l'outil d'aide, expliquer le rôle de la fonction **MappAllocDefault**. En utilisant un point d'arrêt bien placé, donner les valeurs des identifiants "MilApplication", "MilSystem", "MilDisplay". A votre avis, à quoi servent ces valeurs et comment sont elles utilisées ? Afin de vous aider, vous pouvez cliquer sur le mot clé MIL_ID dans la page d'aide de la fonction MappAllocDefault au paragraphe Syntax.
3. En utilisant l'outil d'aide, expliquer le fonctionnement :
 - de la fonction **MbufRestore**. Quel est le rôle de chaque paramètre d'entrée ?
 - de la fonction **MdispSelect**. Quel est le rôle de chaque paramètre d'entrée ?
 - de la fonction **MbufDiskInquire**. Quel est le rôle de chaque paramètre d'entrée ?
 - de la fonction **MbufFree**.
 - de la fonction **MappFreeDefault**.

3. MANIPULATION D'HISTOGRAMME

Nous allons étudier dans cette partie la manipulation d'histogramme afin de rendre binaire une image avant analyse. ***Pour avoir une connaissance de base en traitement d'image, vous devez avant d'aborder cette partie lire attentivement l'annexe 1.***

Afin d'étudier la manipulation d'histogramme, vous devez créer un nouveau projet dans votre espace de travail (Solution) selon les étapes suivantes :

- dans la zone de gestion de projet, sélectionner *Solution 'ImageAnalysis'* puis cliquer sur le bouton droit de la souris et sélectionner : Ajouter -> Nouveau projet... ;
- suivre toutes les étapes décrites dans le paragraphe 1.1.1. Vous pouvez par exemple appeler votre nouveau projet *HistogramAnalysis*. N'oubliez pas de supprimer les fichiers "stdafx.h" et "stdafx.cpp".
- afin de configurer votre nouveau projet pour qu'il puisse utiliser la librairie MIL, vous devez effectuer toutes les étapes décrites dans le paragraphe 1.1.2 ;

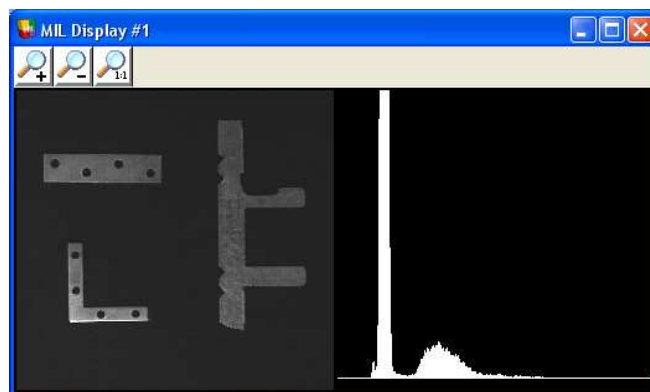
3.1. Rendre le nouveau projet prioritaire

Afin de rendre votre nouveau projet prioritaire, vous devez suivre l'étape suivante :

- dans la zone de gestion de projet, sélectionner votre nouveau projet (dans l'exemple *HistogramAnalysis*) puis cliquer droit avec le bouton de la souris et sélectionner *Définir comme projet de démarrage*.

3.2. Récupérer un code déjà existant

En suivant l'ensemble de la démarche présentée au paragraphe 2.1, copier le code présent dans le fichier "HistogramAnalysis.txt" (à partir du raccourci "sources" présent sur le bureau) dans votre nouveau projet *HistogramAnalysis*. Compiler votre projet et exécuter l'application ainsi obtenue. Vous devez obtenir l'affichage suivant :



3.3. Analyse du code proposé

Afin de vous aider à analyser le code dans son ensemble, vous devez répondre aux questions suivantes :

4. Où est ce qu'est déclarée la fenêtre d'affichage qui permet de visualiser deux images ? Expliquer son fonctionnement.
5. En utilisant l'outil d'aide, expliquer le fonctionnement :
 - de la fonction *MbufChild2d*. Quel est le rôle de chaque paramètre d'entrée ?
 - de la fonction *MbufLoad*. Quel est le rôle de chaque paramètre d'entrée ?
6. A quoi servent les variables *MilLeftSubImage* et *MilRightSubImage* ?
7. Quelles fonctions permettent le calcul et l'affichage de l'histogramme ?

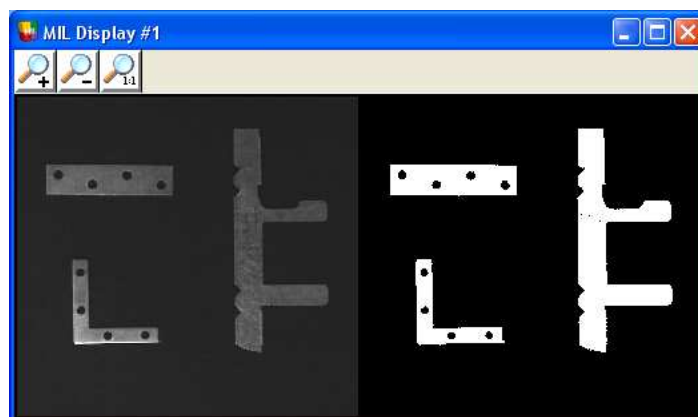
3.4. Manipulation d'un histogramme

6. Interpréter la forme d'histogramme ainsi obtenu. A quoi correspondent les deux bosses ?
7. Modifier votre code afin d'afficher dans la console l'ensemble des valeurs de l'histogramme.
8. A partir de l'étude de ces valeurs, proposer une valeur de seuil qui permettra de séparer les objets du fond de l'image.

3.5. Seuillage (ou binarisation) de l'image

Nous allons utiliser la valeur de seuil ainsi obtenue afin de rendre binaire l'image traitée. L'étape de seuillage est très simple : pour chaque pixel de l'image, il faut tester si la valeur d'intensité du pixel est supérieure à la valeur seuil. Si c'est le cas, on affecte au pixel traité une valeur prédéfinie, sinon on lui affecte la valeur 0.

9. A partir de la documentation et des exemples fournis, quelle fonction appartenant à la librairie MIL proposez-vous d'utiliser ? (afin de vous aider, la fonction à choisir est l'une des fonctions suivantes : *MimMorphic*, *MimBinarize*, *MimClose*, *MimOpen*, *MimLabel*).
10. Compléter votre code afin de binariser l'image initiale en utilisant la valeur seuil obtenue à l'étape précédente. Vous devez afficher le résultat dans la fenêtre de droite. Une fois le code complété, compilé et exécuté, vous devez obtenir le résultat suivant :



4. FILTRES MORPHOLOGIQUES

Nous allons étudier dans cette partie l'utilisation de filtres morphologiques pour l'analyse d'images. Ces méthodes sont très utilisées en industrie car elles ont le mérite d'être simple (d'un point de vue implémentation) et elles sont adaptées aux problématiques de base rencontrées en industrie. Pour ***avoir une connaissance élémentaire sur les filtres morphologique, vous devez avant d'aborder cette partie lire attentivement l'annexe 2 dédié à cet outil.*** Afin d'étudier l'utilisation des filtres morphologiques, vous devez créer un nouveau projet dans votre espace de travail (Solution) selon les étapes suivantes :

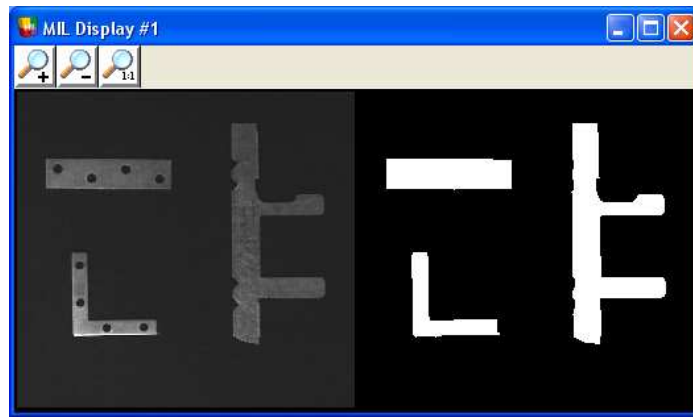
- Dans la zone de gestion de projet, sélectionner *Solution 'ImageAnalysis'* puis cliquer sur le bouton droit de la souris et sélectionner : Ajouter -> Nouveau projet... ;
- Suivre toutes les étapes décrites dans le paragraphe 1.1.1. Vous pouvez par exemple appeler votre nouveau projet *MorphologicalFiltering*. N'oublier pas de supprimer les fichiers "stdafx.h" et "stdafx.cpp".
- Afin de configurer votre nouveau projet pour qu'il puisse utiliser la librairie MIL, vous devez effectuer toutes les étapes décrites dans le paragraphe 1.1.2 ;
- Définir votre nouveau projet comment projet prioritaire (cf. paragraphe 3.1) ;
- Supprimer le code préexistant du fichier "MorphologicalFiltering.cpp" et le remplacer par le code final du projet *HistogramAnalysis* précédant ;
- Compiler puis exécuter votre code afin de tester son bon fonctionnement.

Les filtres morphologiques de bases sont l'érosion, la dilatation, l'ouverture et la fermeture. Ces 4 filtres permettent d'effectuer des tâches élémentaires sur des images binaires ou d'intensité afin de les rendre plus simple pour des étapes d'analyse. Vous trouverez dans l'aide (dans l'onglet *Sommaire* à la rubrique *Image processing*) l'ensemble des informations nécessaires à l'utilisation de ces filtres sous la librairie MIL.

4.1. Méthode de bouchage de trous

Nous allons étudier dans cette partie la possibilité de boucher des trous présents dans un objet.

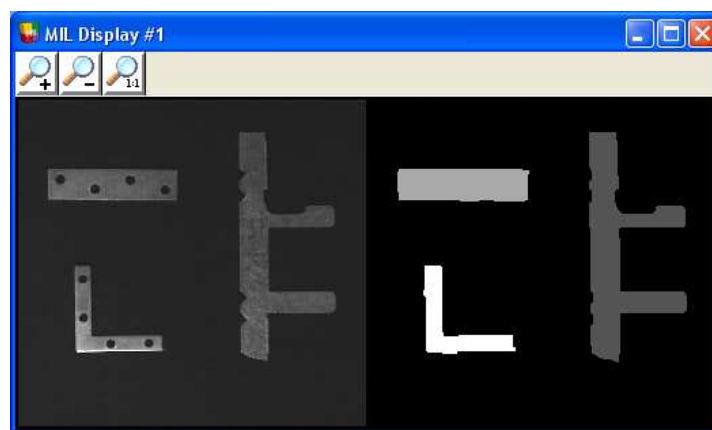
11. A partir de la documentation, donner le nom de la fonction de la librairie MIL qui vous permettra de boucher les trous des deux pièces présentes dans l'image binaire (afin de vous aidez, la fonction à choisir est l'une des fonctions suivantes : *MimMorphic*, *MimBinarize*, *MimClose*, *MimOpen*, *MimLabel*).
12. Quelle taille d'élément structurant permettra d'effectuer un bouchage de trous efficace ?
13. Compléter votre code afin de boucher les trous des deux pièces présentes dans l'image binaire. Vous devez afficher le résultat dans la fenêtre de droite. Une fois le code complété, compilé et exécuté, vous devez obtenir le résultat suivant :



4.2. Méthode de séparation et dénombrement de pièces

Afin de connaître et comprendre les fonctions permettant la séparation et le dénombrement de pièces sous la librairie MIL, vous devez étudier l'exemple *mimprocessing* fourni dans le programme "Example Launcher" (accessible à partir du raccourci "Mil Control Center" présent sur le bureau). Afin de vous aider à analyser le code dans son ensemble, vous devez répondre aux questions suivantes :

14. Quelle fonction permet de séparer (ou labéliser) les différents objets dans une image ?
15. Comment est construite l'image de sortie (ou de destination) ?
16. Quel facteur de multiplication faut-il appliquer à l'image de sortie pour que la valeur d'intensité maximale soit égale à 256 ?
17. Compléter votre code afin d'afficher une image où chaque pièce est différenciée par une intensité différente. Vous devez afficher le résultat dans la fenêtre de droite. Une fois le code complété, compilé et exécuté, vous devez obtenir le résultat suivant :



18. Quelles fonctions permettent d'extraire le nombre total d'objet présent dans l'image ?
19. Compléter votre code afin d'afficher dans la fenêtre de dialogue le nombre de pièces distinctes présentes dans l'image et détectées de façon automatique. Ici la valeur obtenue sera bien évidemment 3.

5. METROLOGIE : METHODE DE MESURE DE CARACTERISTIQUES D'OBJETS

Nous allons étudier dans cette dernière partie un exemple de manipulation d'outil de métrologie afin de calculer le périmètre de chaque objet détecté. Pour cela, vous devez créer un nouveau projet dans votre espace de travail (Solution) selon les étapes suivantes :

- Dans la zone de gestion de projet, sélectionner *Solution 'ImageAnalysis'* puis cliquer sur le bouton droit de la souris et sélectionner : Ajouter -> Nouveau projet... ;
- Suivre toutes les étapes décrites dans le paragraphe 1.1.1. Vous pouvez par exemple appeler votre nouveau projet *Metrology*. N'oublier pas de supprimer les fichiers "stdafx.h" et "stdafx.cpp".
- Afin de configurer votre nouveau projet pour qu'il puisse utiliser la librairie MIL, vous devez effectuer toutes les étapes décrites dans le paragraphe 1.1.2 ; Attention, vous devez inclure en plus la librairie intitulée : "miledge.lib" à votre projet ;
- Définir votre nouveau projet comment projet prioritaire (cf. paragraphe 3.1) ;
- Supprimer le code préexistant du fichier "Metrology.cpp" et le remplacer par le code présent dans le fichier texte "Metrology.txt" (fichier accessible à partir du raccourci "sources" présent sur le bureau) ;
- Compiler puis exécuter votre code afin de tester son bon fonctionnement.

5.1. Méthode de détection de contours

Afin de connaître et comprendre les fonctions permettant la détection de contour sous la librairie MIL, vous devez étudier l'exemple *medge* fourni dans le programme "Example Launcher" (accessible à partir du raccourci "Mil Control Center" présent sur le bureau).

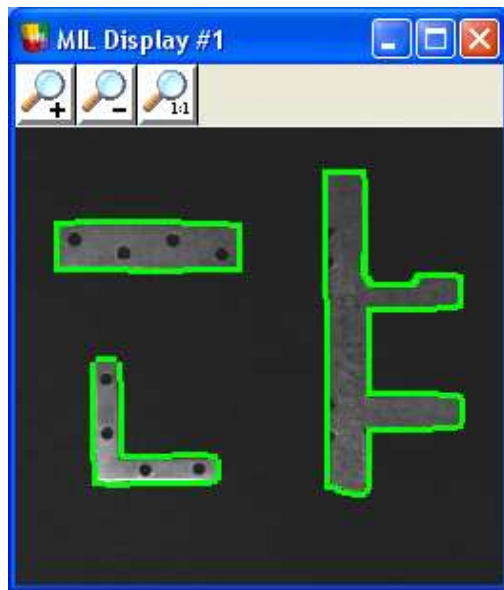
Afin de vous aider à analyser le code dans son ensemble, vous devez répondre aux questions suivantes :

20. A quoi servent les fonctions *MedgeAlloc* et *MedgeAllocResult* ?
21. Quel est le rôle de la fonction *MedgeControl* ? En utilisant l'aide, donner le paramètre d'entrée qu'il faut fournir à cette fonction pour calculer le périmètre des contours extraits ?
22. A quoi sert la fonction *MedgeCalculate* ? Quelles résultats renvoie t'elle et dans quelle variable est-ce stocké ?
23. Quelles fonctions permettent de tracer en surbrillance les contours des objets extraits ?
24. Compléter votre code afin d'afficher une image où chaque contour externe d'objet est tracé en surbrillance. Vous devez afficher le résultat dans la seule fenêtre. Une fois le code complété, compilé et exécuté, vous devez obtenir le résultat de la figure 2a.
25. Compléter votre code afin d'afficher un numéro pour chaque contour détecté. Vous devez afficher le résultat dans la seule fenêtre. Une fois le code complété, compilé et exécuté, vous devez obtenir le résultat de la figure 2b.

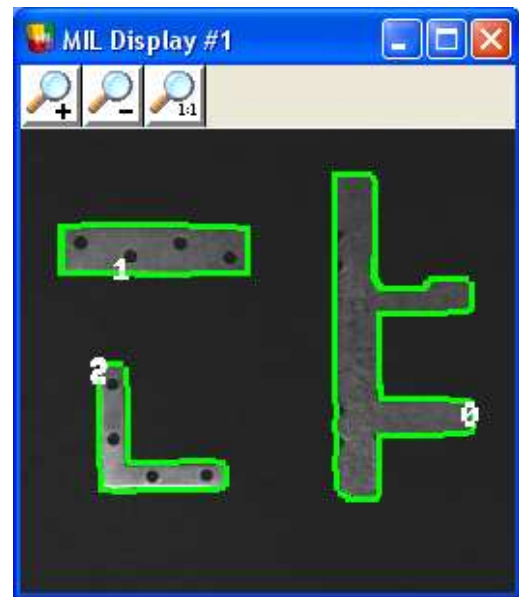
Remarque : si vous avez des difficultés à détecter trois contours, cela provient de la méthode de détection de contours utilisé par défaut. Afin de rendre la méthode plus robuste, vous pouvez jouer sur le paramètre de lissage de la méthode en ajoutant dans votre code la ligne de programmation suivante :

```
MedgeControl( MilEdgeContext, M_FILTER_SMOOTHNESS, 80.0 ) ;
```

26. Compléter votre code afin d'afficher dans la fenêtre de dialogue le résultat du calcul du périmètre de chaque objet détecté. Dans l'exemple de la figure 2b, vous devez trouver des valeurs voisines de 557 pixels pour l'objet 0, 228 pixels pour l'objet 1 et 246 pixels pour l'objet 2.



a – contours détectés

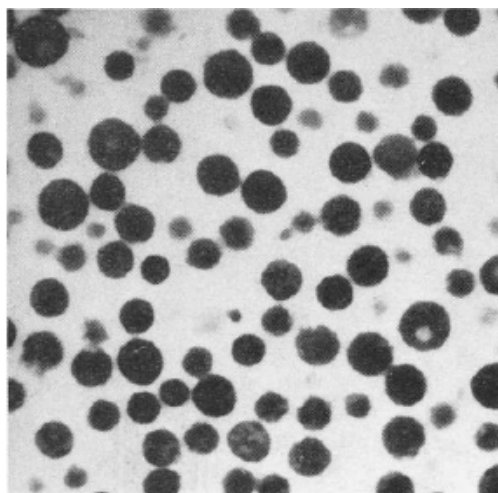


b – contours numérés et analysé

Figure 2 : résultats à obtenir dans la partie métrologie

6. CHALLENGE

Maintenant que vous maîtrisez la librairie MIL, proposer et coder une chaîne de traitement permettant d'extraire toutes les cellules biologiques de l'image ci-dessous ainsi que de calculer de façon automatique l'aire de chaque cellule extraite. Vous pouvez récupérer l'image à partir du chemin suivant : "C:\TPVISION\Images\finalImage.mim"



ANNEXES

ANNEXE 1 : METHODES EN TRAITEMENT D'IMAGES NUMERIQUES

Source : Cours en traitement d'image numérique – INSA de Lyon - département génie électrique – option TDSI, Technique de l'ingénieur : analyse d'images (<http://www.techniques-ingenieur.fr>)

À chaque application correspond un programme d'analyse d'images, qui se déroule toujours de la même façon. Un programme d'analyse d'images effectue une suite d'instructions, qui correspond à une décomposition logique du problème posé. Les étapes sont visualisées sur l'écran de l'analyseur d'images (*figure 1*) et permettent de suivre le déroulement du programme. Un programme comporte quatre étapes essentielles :

- acquisition de l'image ;
- sélection des zones d'intérêt ;
- mesures ;
- interprétation des résultats.



Figure1 : Analyseur d'images Quantimet 600 de Leica

6.1. Acquisition

L'analyse d'images permet l'étude de nombreuses images, le dispositif d'entrée étant adapté à chaque situation pour acquérir l'image. Les analyseurs peuvent être reliés à des caméras industrielles (cf. annexe **partie 3 intitulée «Acquisition d'une image numérique par caméras CCD »**), des microscopes optiques (à transmission, à réflexion ou inversés) (*figure 1*), ou des microscopes électroniques (à balayage ou à transmission). Ils permettent, une caméra étant en entrée, de numériser des documents photographiques (négatifs ou positifs), des clichés radiologiques et toutes sortes d'objets (pièces de montage, circuit intégré, structure microscopique de matériaux,...).

La première étape est celle qui permet au système d'acquérir les images représentatives des objets à analyser. À une image, vont être associés un certain nombre de points images, et à chaque point une (ou plusieurs) valeur(s) représentative(s) de l'image en ce point.

Acquisition monochrome

L'acquisition d'une image monochrome consiste dans l'enregistrement d'une image numérique (ou de gris), où à l'image est associé un certain nombre de points images (le plus souvent 512x512), et à chaque point une valeur représentative du niveau de gris de l'image en ce point (le plus souvent compris entre 0 et 255 = 256 niveaux de gris).

L'acquisition de l'image est réalisée le plus souvent à l'aide d'une caméra à tube ou à transfert de charge (CCD : *Charge Coupled Device*, cf. annexe **XX**). Les caméras à transfert de charge sont les plus répandues, les caméras vidéo sont les plus anciennes et tendent à disparaître.

Les performances d'une caméra sont définies par sa résolution spatiale, sa vitesse d'acquisition, sa possibilité de réponse dans des conditions d'éclairement plus ou moins difficiles (sensibilité), et dans différentes gammes de longueurs d'onde, sa capacité à discriminer un certain nombre de niveaux de gris, et la forme de la réponse, notamment la linéarité et la dynamique (gamme possible de densités optiques). Les caméras à transfert de charge ont une bonne sensibilité et une bonne linéarité. Les caméras à tube, principalement à tube vidicon, présentent une bonne dynamique. Les performances réelles obtenues vont dépendre de l'ensemble de la chaîne d'acquisition, notamment des caractéristiques de la carte d'acquisition et l'ensemble doit être cohérent.

Pour des applications à bas niveaux de lumière comme en fluorescence ou en microscopie électronique à transmission (TEM), on utilise des caméras particulières. Les caméras CCD avec intégration, le plus souvent refroidies, permettent d'intégrer la luminosité sur des temps plus ou moins longs. Elles autorisent une bonne résolution, mais nécessitent des applications statiques. Les caméras CCD intensifiées, les caméras à tube SIT (Silicon Intensifier Target) ou ISIT (Intensified SIT) permettent une acquisition rapide et sont recommandées pour les applications dynamiques nécessitant des temps d'exposition très courts.

Acquisition couleur

La couleur intervient déjà dans l'acquisition monochrome, par le jeu de filtres colorés qui permettent d'augmenter les contrastes. L'acquisition directe d'une image couleur est obtenue grâce à une caméra couleur (tri CCD par exemple). Elle consiste dans l'enregistrement de trois images numériques correspondant aux trois couleurs primaires additives : rouge, vert, bleu.

Transfert d'une image numérique

Les logiciels d'analyse d'images disposent tous d'une entrée pour des images numériques, sous un ou plusieurs formats informatiques, comme les formats TIFF, BMP, PNG ou JPG. Cette possibilité permet d'analyser en temps différé des images numériques issues de systèmes d'acquisition comme une caméra CCD qui acquiert une image et la diffuse par réseau filaire ou sans fils.

6.2. Représentation d'une image numérique

Nous nous intéressons plus particulièrement dans cette partie à l'acquisition et l'analyse d'images monochromes. Une fois une telle image acquise, on peut la représenter comme une matrice de dimension $[dimI, dimJ]$ où chaque composante correspond à un pixel de coordonnées (x,y) et où chaque valeur correspond à l'intensité d'un pixel associée. Figure 2 donne une illustration d'une telle représentation.

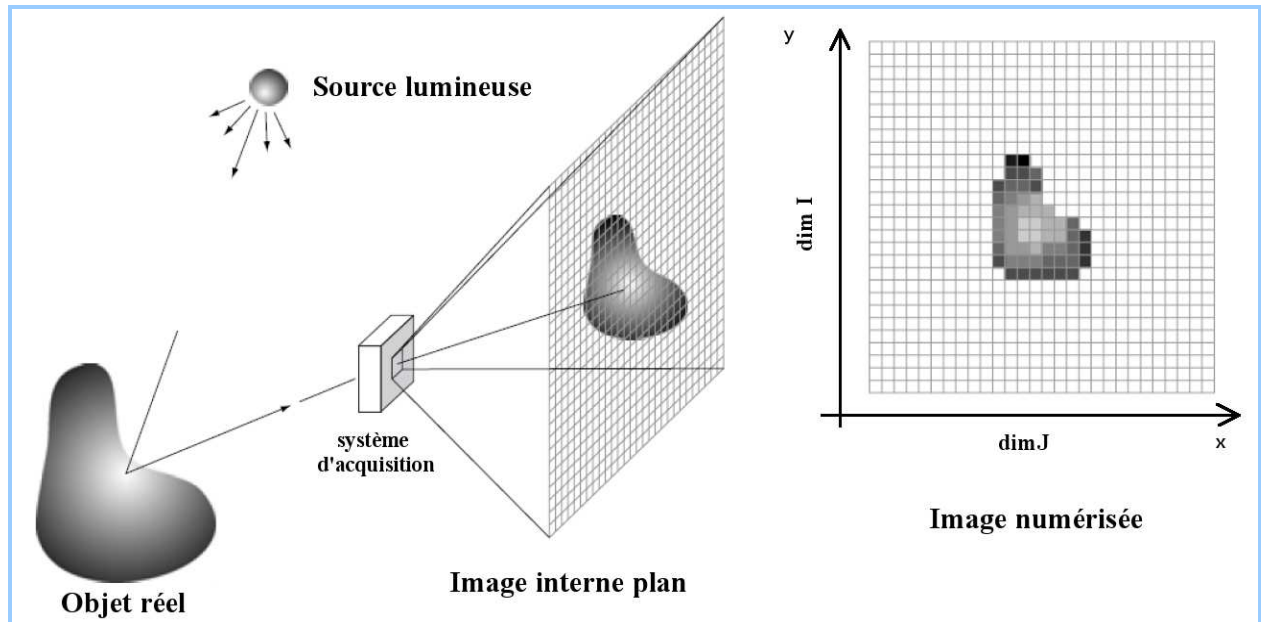


Figure 2 : Illustration de la représentation des images numériques

6.3. Transformations d'images

La phase d'acquisition a permis d'associer à une image un ensemble de données. Des logiciels vont permettre d'améliorer l'image, puis de la décomposer, de la segmenter en plusieurs phases (composants de même nature), afin de sélectionner la ou les régions d'intérêt, en vue d'une quantification ou d'une reconnaissance de formes. L'image initiale (figure 3a) est en général traitée, transformée en une autre image numérique, puis seuillée pour donner une image binaire (figure 3b), souvent encore transformée en d'autres images binaires (figure 3c et d). À chaque étape, des comparaisons et des opérations entre images numériques ou binaires peuvent se faire. Cette étape de l'analyse est souvent complexe.

6.3.1. Opérations entre images

Deux images peuvent être mises en correspondance point à point, et une opération peut s'effectuer entre ces deux images pour créer une troisième image.

Opérations entre images numériques

Deux images numériques peuvent être mises en correspondance, et des opérations arithmétiques ou des opérations logiques peuvent être définies. La moyenne d'une série d'images successives d'un objet statique est une transformation efficace pour réduire le bruit (le plus souvent gaussien) et améliorer l'image.

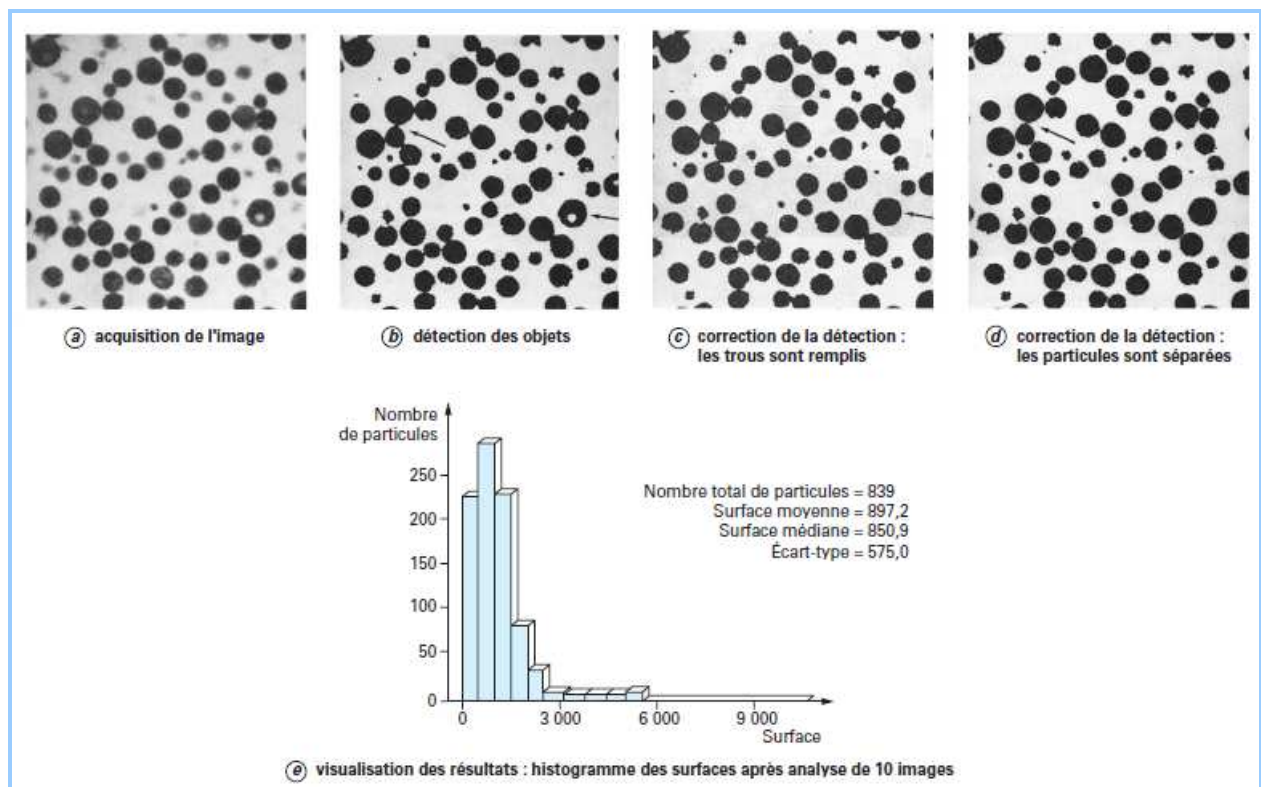


Figure 3 : Exemple de programme : granulométrie en nombre

Des opérations entre les trois images : rouge, vert, bleu résultant de l'acquisition couleur permettent de passer de l'espace trichromatique : rouge, vert, bleu, à l'espace tridimensionnel de teinte, luminance, et saturation. La teinte reflète la couleur (longueur d'onde) principale. La luminance correspond à l'image noir et blanc, obtenue sans filtres colorés. La saturation donne le degré de pureté d'un point par rapport à sa couleur primaire principale. Plusieurs modèles colorimétriques existent, et de nombreuses combinaisons des trois images initiales peuvent être utilisées pour segmenter l'image.

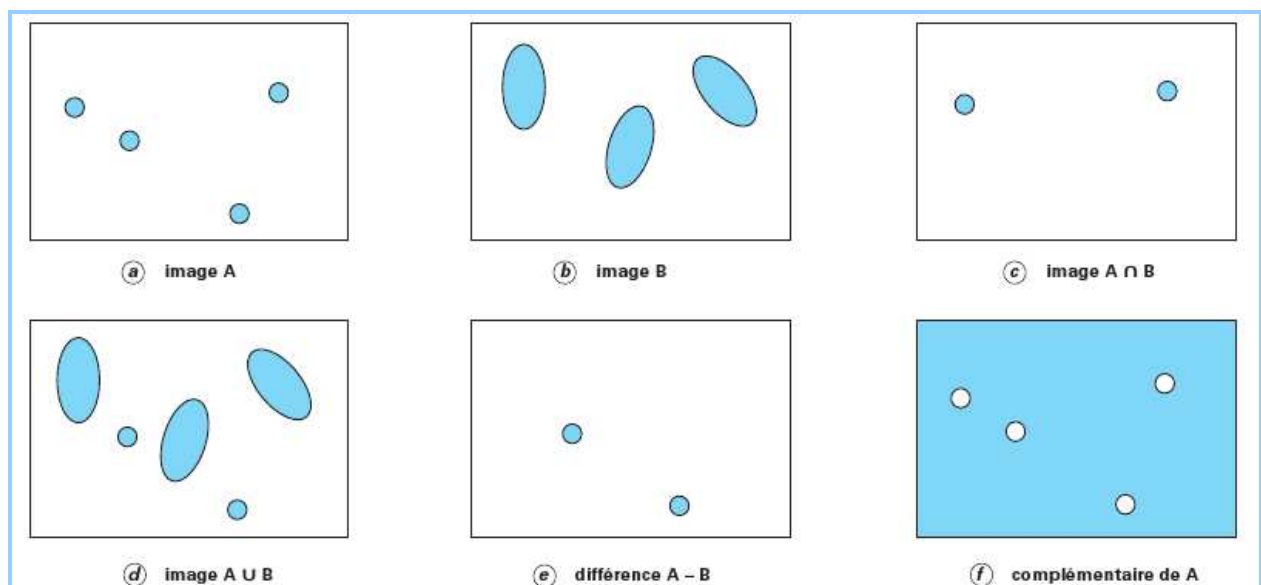


Figure 4 : Opérations entre images binaires

Opérations entre images binaires

Les opérations booléennes consistent à mettre en correspondance, point à point, deux images binaires et à obtenir l'image : intersection, union, différence de deux images (figure 4).

6.3.2. Transformations point par point

Tous les points d'une image peuvent être transformés, point par point, en fonction de règles, et chacun indépendamment des autres points de l'image.

Modification d'histogrammes

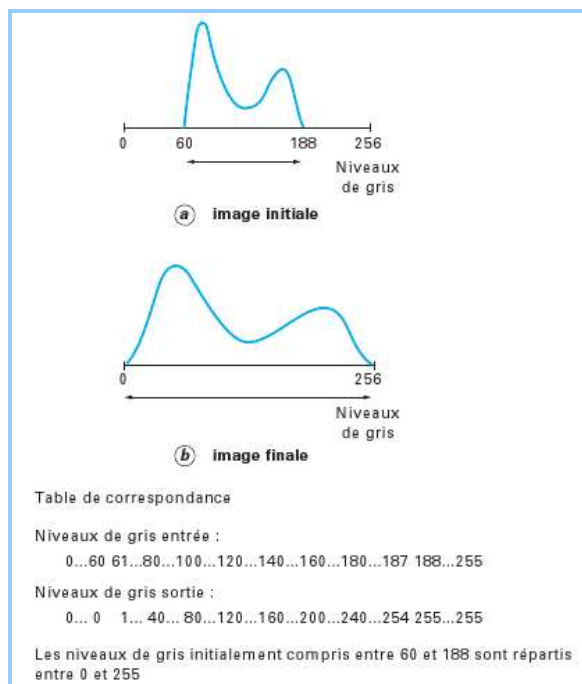


Figure 5 : Transformation point par point : augmentation de contraste

On peut inverser une image numérique et passer rapidement du négatif au positif. On peut changer la gamme de niveaux de gris, associer aux valeurs extrêmes théoriques (0-255 par exemple) les valeurs extrêmes observées (minimum et maximum), étaler ainsi les niveaux de gris de l'image et améliorer la visualisation de l'image (figure 5). On peut calibrer les niveaux de gris en fonction d'une gamme étalon. La correspondance entre les niveaux de gris d'entrée et de sortie s'effectue rapidement par des tables de correspondance. Dans une transformation d'extension des niveaux de gris (figure 5), l'opération :

$$\text{int}[(x_i - x_{\min}) \times 256 / (x_{\max} - x_{\min})]$$

int désignant la partie entière, n'est réalisée pour les 256 valeurs qu'une seule fois à la mise en place de la table de correspondance.

Seuillage

Le seuillage est aussi une transformation point par point. Il permet de passer d'une image numérique (niveaux de gris) à une image binaire (noir et blanc). Il permet de ne retenir dans l'image que les points, qui ont un niveau de gris supérieur ou inférieur à un seuil donné (points les plus sombres, ou points les plus clairs), ou ceux qui sont compris entre deux valeurs de gris $G_1 - G_2$ (phase grise). Dans notre exemple de granulométrie (figures 3a et b), les points les plus sombres sont retenus. Les échantillons doivent toujours être préparés dans des conditions qui optimisent les contrastes. Une mise au point des techniques préparatives pour l'analyse d'images est souvent obligatoire.

6.3.3. Filtres linéaires par convolution

Tous les points d'une image sont entourés de voisins plus ou moins proches. Dans une transformation locale, la valeur d'un point est définie en fonction de sa valeur et de celle de ses voisins, et dans une transformation linéaire, l'image est le résultat du produit de convolution de l'image initiale par un filtre linéaire de taille 3x3 ou plus (figure 7).

Filtres passe-bas

Le filtre le plus simple, *le filtre uniforme*, consiste à remplacer la valeur de chaque point par la valeur moyenne effectuée avec ses voisins. C'est un filtre passe-bas, qui élimine les petites fluctuations de l'image, réduit le bruit mais peut cependant modifier les petits détails de l'image. Les conséquences néfastes d'un tel filtre ressortent nettement sur une image test de fort contraste et comportant des contours nets (figure 8a), sur laquelle un filtre uniforme de taille a été appliqué (figure 8b).

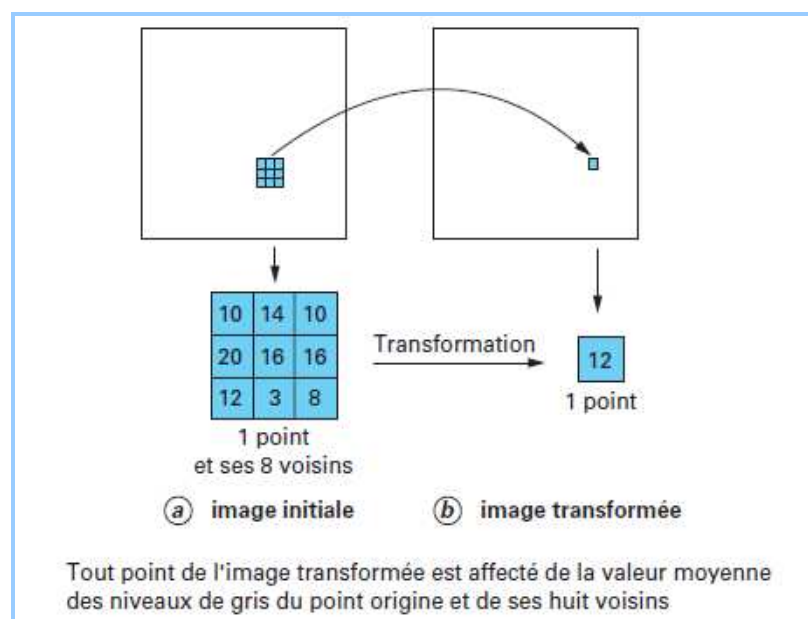


Figure 7 : Transformation locale linéaire : exemple d'un filtre uniforme de taille 3x3

6.4. Mesures

Les méthodes de segmentation ont permis d'isoler dans l'image les objets à quantifier ou à reconnaître. Les mesures possibles sont très nombreuses et dépendent du problème posé. Les objectifs doivent être bien définis au préalable, pour rendre l'analyse efficace.

6.4.1. Fenêtre de mesure

Le champ de mesure ne correspond pas obligatoirement à l'image toute entière. L'utilisateur décide de sa forme, de sa taille et de sa position dans l'image. Il peut programmer son déplacement, de même que celui d'une platine porte-échantillon et d'un système de focalisation automatique. Il peut analyser une image par secteurs, ou au contraire un ensemble d'images.

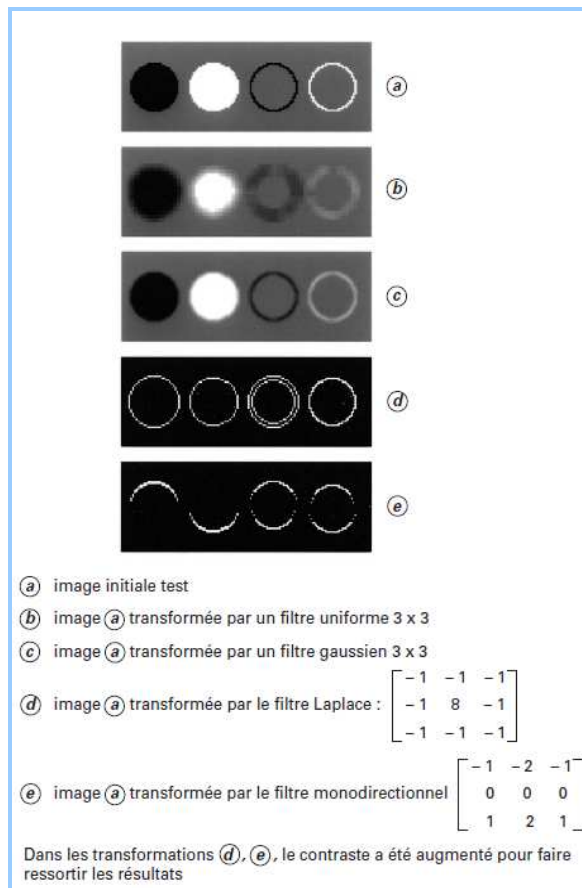


Figure 8 : Images de transformations linéaires

6.4.2. Paramètres de forme

Les différentes variables accessibles dans un analyseur d'images, associées à toutes les transformations d'images possibles, permettent de rendre compte de beaucoup de formes. L'indice de circularité (ou rondeur) est défini par le rapport du périmètre au carré sur la surface, divisé par la constante 4π . Il est très rapide d'acquisition. Dans le cas d'un rond, il est minimal et est égal à 1, et il s'écarte de cette valeur lors d'une élongation, d'une invagination ou en présence de trous (figure 9).

Dans ces cas, le paramètre d'élongation : rapport du diamètre de Férêt maximal au diamètre de Férêt minimal, convient bien, malgré un temps de mesure plus long. Le diamètre de Férêt se définit comme la distance séparant les deux tangentes parallèles à une direction donnée et qui encadrent la particule (figures 9 et 10).

D'autres paramètres de forme peuvent être définis, par exemple

- le rapport de la projection sur le diamètre de Férêt (figure 10) ;
- le rapport du périmètre sur le périmètre convexe ;
- le rapport de la surface du plus grand cercle circonscrit sur la surface du plus grand cercle inclus.

De façon générale, le filtre gaussien ou de moyenne pondérée (figure 8c) ou, mieux, le filtre non linéaire de la médiane donnent des résultats plus satisfaisants que le filtre uniforme pour réduire le bruit d'une image. Ce sont des méthodes à utiliser en complément de la procédure d'acquisition d'images successives ou lorsqu'elle n'est pas possible (images dynamiques par exemple).

Filtres passe-haut

Dans un filtrage passe-haut, les coefficients associés aux points sont choisis pour mettre en valeur, au contraire du filtrage passe-bas, des contours. Deux exemples de transformations sont montrés sur l'image test : un filtre de type Laplace, et un autre monodirectionnel (figures 8d et e). Une transformation point par point d'augmentation de contraste peut être nécessaire pour visualiser le résultat de la transformation.

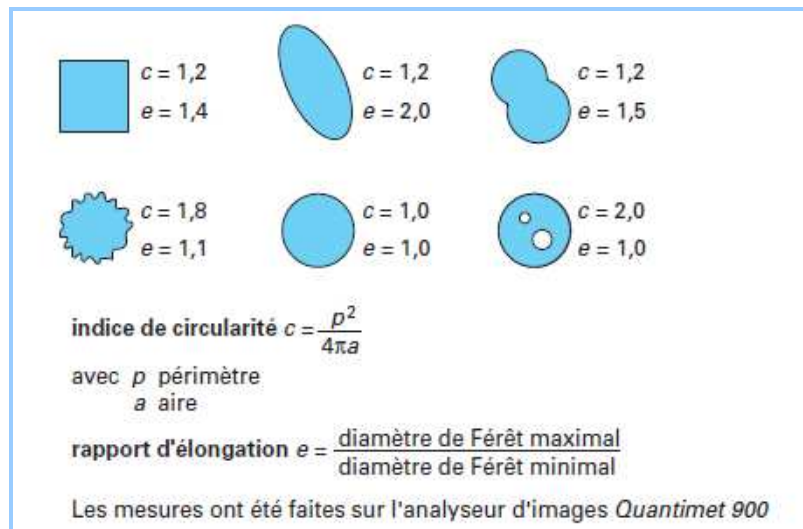


Figure 9: Indices de circularité et d'élongation

Ces derniers exemples montrent l'intérêt des transformations comme paramètres de mesure et non plus seulement pour sélectionner des zones. La réciproque est aussi vraie. Le paramètre de rondeur est souvent utilisé pour identifier les objets isolés qui seront ensuite analysés. D'autres approches basées sur la transformée de Fourier sont aussi employées.

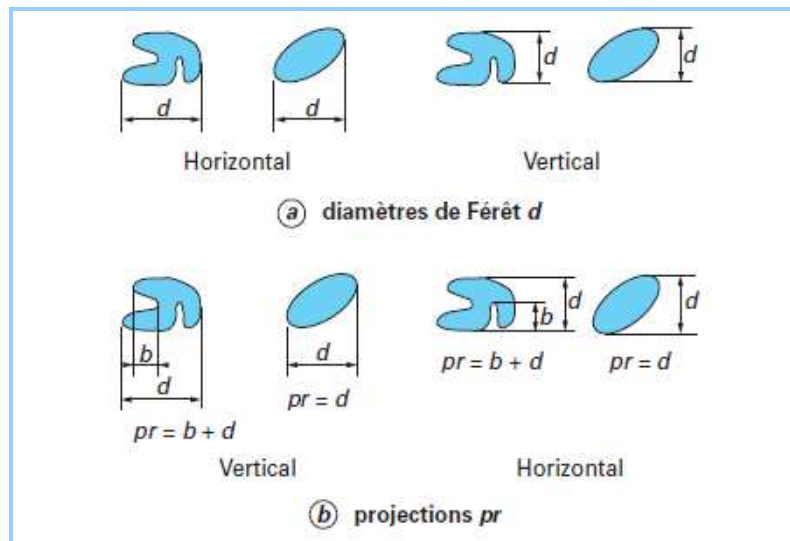


Figure 10 : Diamètre de Férêt et projections

6.4.3. Paramètres de texture

La texture se définit comme l'étude de la distribution des différents niveaux de gris, à l'intérieur d'une structure. Un premier groupe de méthodes consiste à étudier la distribution des différents niveaux de gris, présents dans la structure, sans tenir compte de leur situation topographique. L'information est résumée par des variables comme le coefficient de variation, de dissymétrie ou d'aplatissement, ou l'importance des points les plus noirs ou les plus blancs. Ces mesures peuvent être effectuées dans différentes parties de la structure, par exemple, du centre à la périphérie pour une mesure d'excentricité. De même, l'étude des points les plus clairs ou les plus sombres peut se compléter par l'analyse des images de ces points extrêmes : évaluation du pourcentage de points rassemblés en amas ou de l'excentricité par exemple. La comparaison

entre un point et ses voisins a donné naissance à d'autres développements. Les matrices de co-occurrence ou les matrices de longueurs de section en sont des exemples. Dans le premier cas, les variables sont obtenues à partir du dénombrement des couples de points de niveaux de gris (i, j) , les niveaux de gris étant regroupés en classes de niveaux de gris. Dans la méthode de longueurs de sections, les variables sont obtenues à partir de la longueur des suites de points de même classe de niveau de gris i .

ANNEXE 2 : FILTRES DE MORPHOLOGIE MATHÉMATIQUE

Source : Technique de l'ingénieur : analyse d'images (<http://www.techniques-ingenieur.fr>)

Les transformations de morphologie mathématique sont des transformations locales. Ce sont des transformations non linéaires, d'origine ensembliste. Elles font intervenir un élément particulier, que l'on désigne sous le terme d'élément structurant. Cet élément se définit par sa taille, sa forme, sa direction éventuellement, et son point d'origine. Cela peut être un cercle de rayon r et dont l'origine se situerait au centre, ou un segment horizontal de longueur l avec comme origine une extrémité, ou encore un ensemble de bipoints séparés par un espace de n points.

6.5. Principe de filtrage en morphologie mathématique

Le principe de filtrage en morphologie mathématique est très simple. L'idée de base est de faire parcourir un élément structurant localisé par son centre sur toute l'image initiale. Pour chaque position de son centre, on effectue une opération non-linéaire qui fait intervenir les pixels de l'image initiale inclus dans l'élément structurant. Le résultat de l'opération est affecté au pixel de l'image transformée ayant pour coordonnées, les coordonnées du centre de l'élément structurant. Figure 1 donne un exemple d'application d'un tel filtre pour un élément structurant ayant pour forme une croix.

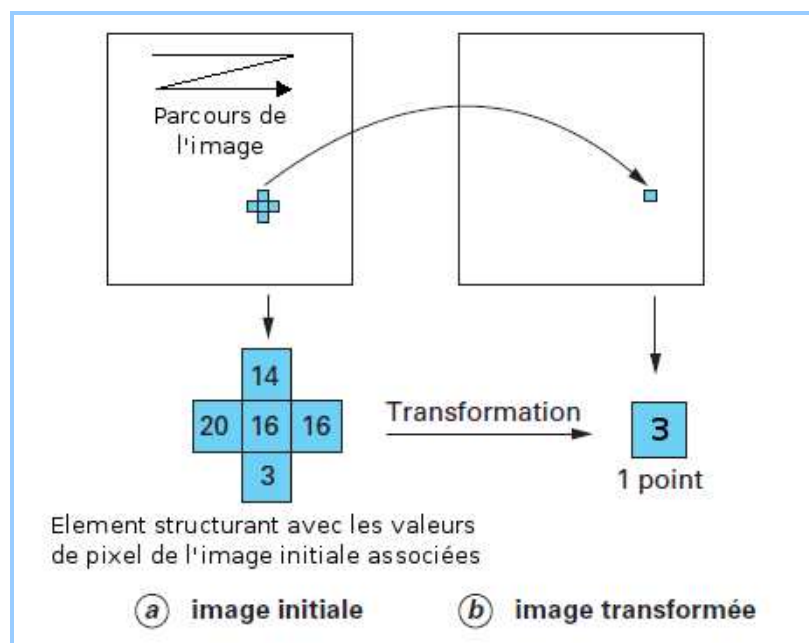


Figure 1 : Illustration du principe de filtrage en morphologie mathématique

Il est à noter que la taille et la forme de l'élément structurant influe directement sur la qualité des résultats obtenus. De plus le choix de la taille de l'élément structurant peut permettre de mettre en évidence des structures de l'image à des échelles différentes.

6.6. Erosion et dilatation

L'**érosion** d'une image binaire ou d'une image numérique par un élément structurant plan est l'opération qui permet de passer d'une image à une image érodée, telle qu'à tout point de l'image, en correspondance avec l'origine de l'élément structurant, soit associée la valeur minimale des points en correspondance avec tous les points de l'élément structurant (figures 2b et 3c). Sur une image binaire, l'érodé d'une image X est l'ensemble des points tel que l'élément structurant est inclus dans X.

La **dilatation** est l'opération qui associe, à tout point de l'image, la valeur maximale des points en correspondance avec tous les points de l'élément structurant (figures 2c et 3d). Le dilaté d'une image binaire est l'ensemble des points tel qu'il existe une intersection non nulle entre l'élément structurant et l'image initiale. N'importe quelle suite d'érosions et de dilatations peut présenter un intérêt et transformer efficacement une image.

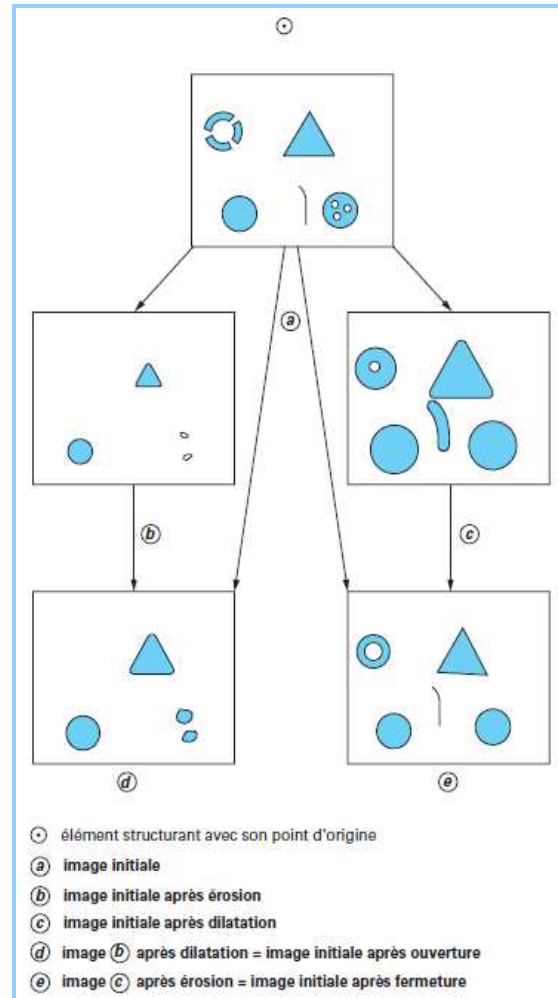


Figure 2 : Transformations binaires de morphologie mathématique

6.7. Ouverture et fermeture

Une **ouverture** est le terme qui définit **une érosion** par un élément structurant **suivie d'une dilatation** par le transposé de l'élément structurant initial. Sur une image binaire, l'ouvert est l'ensemble formé par tous les éléments structurants contenus dans l'image initiale. Une ouverture fait disparaître les éléments de largeur inférieure à celle de l'élément structurant, et nettoie efficacement l'image de petits points parasites (le bruit). Elle a un rôle de tamis (figures 2d et 5). Sur une image numérique, une ouverture arase les maximums (figure 2e) et élimine les fines zones brillantes (figure 3b). L'élément structurant détermine la taille des détails ôtés.

La **fermeture** est le terme qui définit **une dilatation** par un élément structurant **suivie d'une érosion** par l'élément structurant transposé. Sur une image binaire, une fermeture permet de rapprocher des points, de combler des trous (figures 2e). Sur une image numérique, une fermeture comble les minimums (figure 3f) et élimine les détails fins sombres (figure 4c)

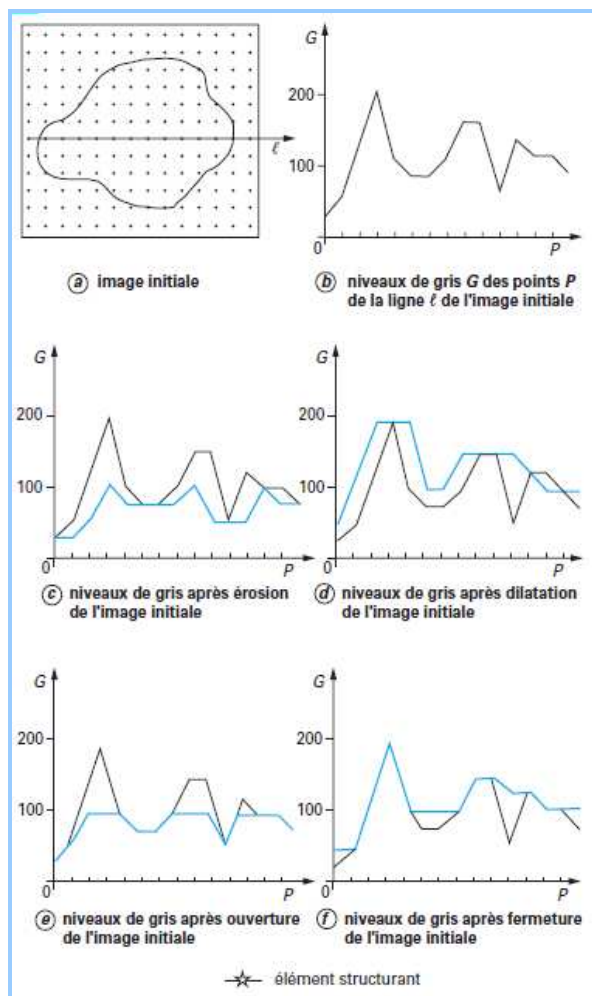


Figure 3 : Transformation de morphologie mathématique en niveau de gris

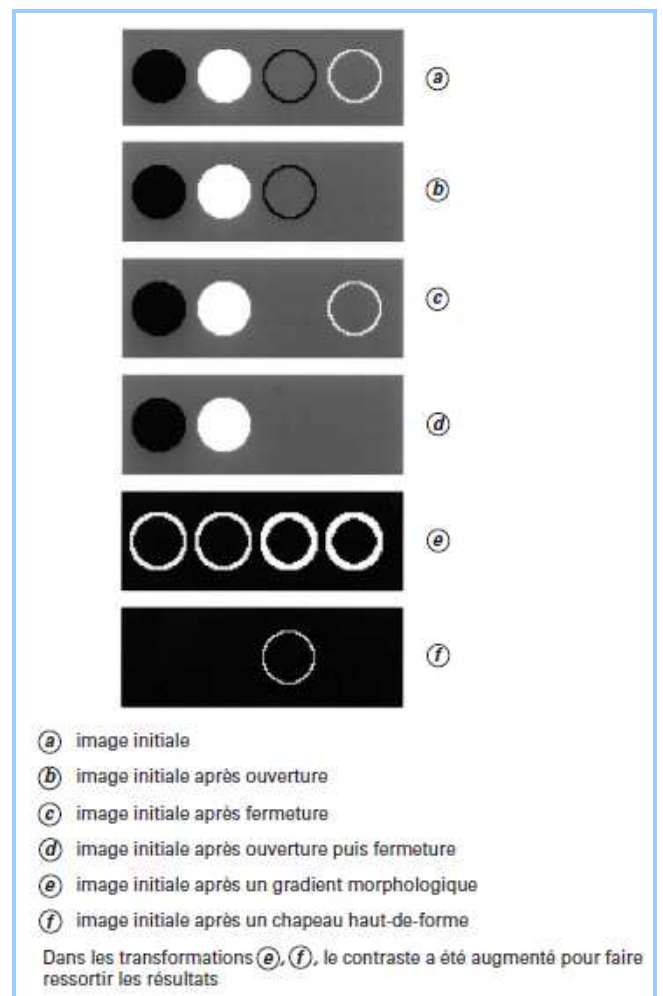


Figure 4 : Images de transformations de morphologie mathématique

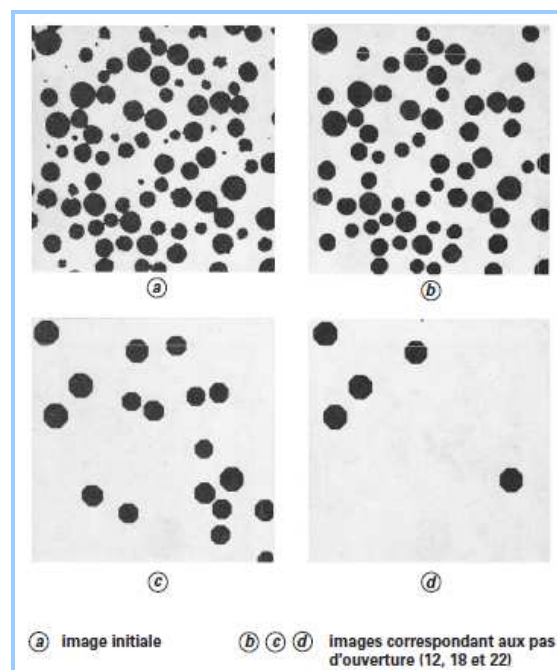


Figure 5 : Granulométrie en mesure

6.8. Squelette

Le **squelette** est une transformation obtenue par **amincissement des éléments connexes de l'image**, à partir des bords, jusqu'à obtenir « l'ossature » ou l'axe médian des structures (figure 6). Le squelette est l'ensemble des points qui se situent à égale distance de deux bords. Il peut se définir aussi comme l'ensemble des centres des boules maximales contenues dans l'objet. Les points particuliers du squelette, comme les points extrêmes ou les points de croisements, sont utiles comme marqueurs ou pour définir une distance entre les extrémités d'une courbe.

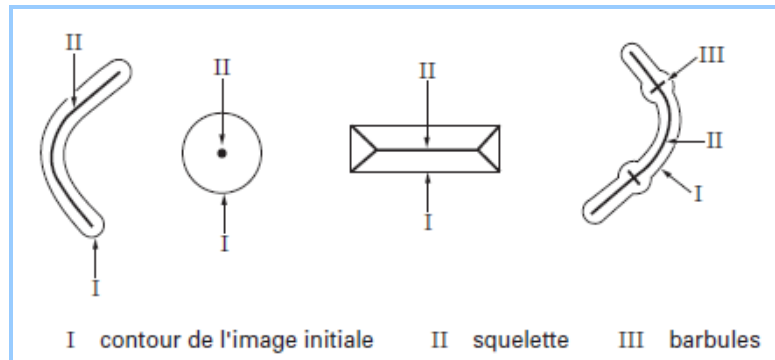


Figure 6 : Transformation de squelette

6.9. Transformations géodésiques

Des transformations peuvent s'effectuer conditionnellement à un autre ensemble : transformations conditionnelles, ou selon la distance géodésique (plus court chemin à l'intérieur d'un ensemble) : transformations géodésiques. Une **dilatation conditionnelle** correspond à une dilatation de taille n suivie d'une intersection avec l'ensemble de référence. Une **dilatation géodésique** correspond à la répétition n fois d'une dilatation de taille élémentaire suivie d'une intersection avec l'ensemble de référence. Un exemple emprunté à la morphométrie osseuse (figure 7) montre le résultat et les étapes d'une dilatation géodésique : dilatation de l'ensemble noir par rapport à l'ensemble noir et blanc. La transformation a été nécessaire pour quantifier l'importance des zones noires et des zones blanches contiguës aux zones noires. La dilatation géodésique est souvent utilisée pour reconstruire une structure à partir d'un marqueur.

Des suites de transformations permettent de transformer efficacement des images. Une suite alternée d'ouvertures et de fermetures numériques de taille croissante permet d'ôter tous les détails fins, sombres et clairs (figure 4d). La différence entre l'image dilatée et l'image érodée donne une image gradient (figure 4e). Le deuxième exemple de la granulométrie (figure 8) montre aussi l'intérêt d'une suite de transformations et d'opérations entre images. Le remplissage des trous (figure 8c) s'effectue par reconstruction géodésique de l'image inverse à partir des bords de l'image, puis inversion de l'image résultat. La procédure utilisée pour la séparation automatique de particules en amas (figure 8d) comporte une reconstruction géodésique des particules à partir de l'image des érodés ultimes (figure 9).

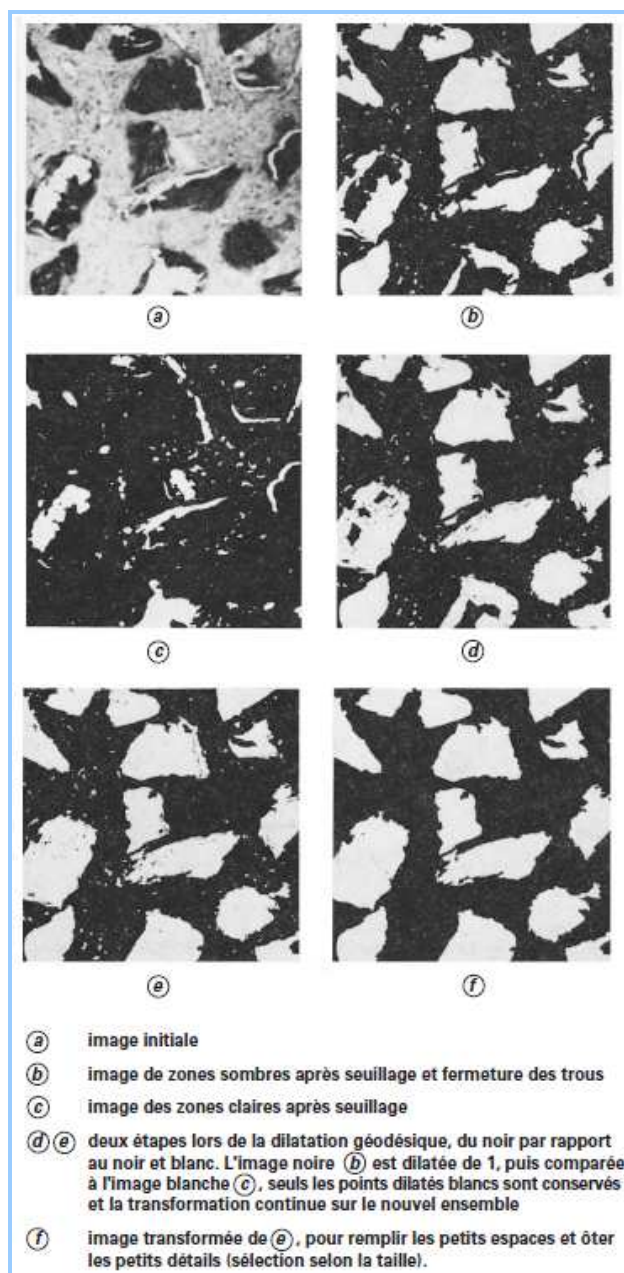


Figure 7 : Exemple de dilatation géodésique

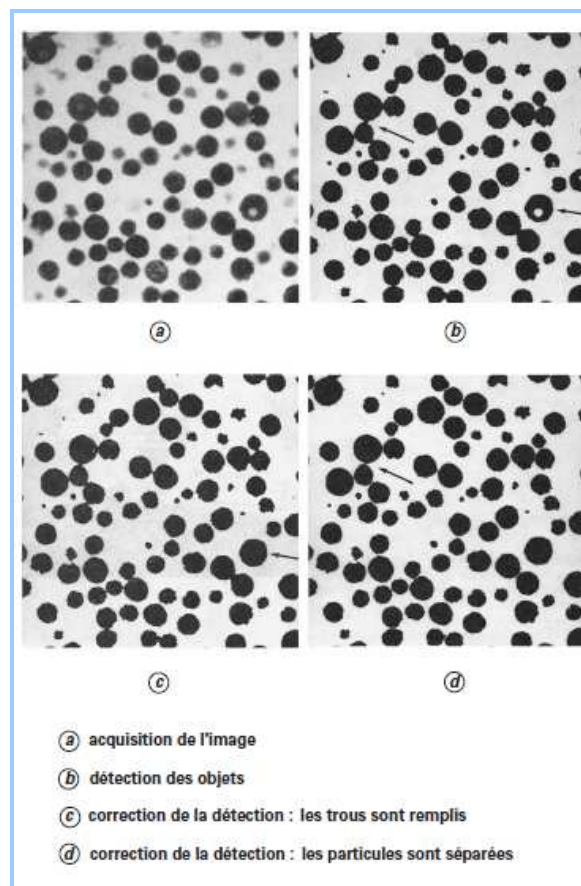
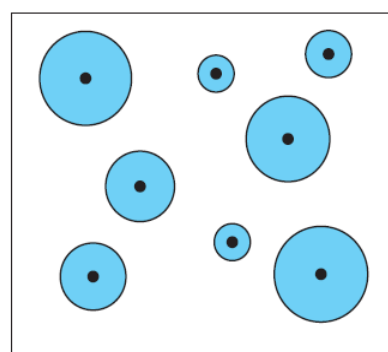


Figure 8 : Exemple de traitement complet en granulométrie



Les érodés sont représentés par les points au centre des images

Figure 9 : Erodés ultimes